

(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau



(43) International Publication Date
14 June 2001 (14.06.2001)

PCT

(10) International Publication Number
WO 01/42949 A2

(51) International Patent Classification⁷: **G06F 17/00**

(21) International Application Number: **PCT/GB00/04180**

(22) International Filing Date: **31 October 2000 (31.10.2000)**

(25) Filing Language: **English**

(26) Publication Language: **English**

(30) Priority Data:
60/169,620 8 December 1999 (08.12.1999) US

(71) Applicant (*for all designated States except US*): **BROKER-TO-BROKER NETWORKS, INC** [US/US]; 1209 Orange Street, City of Washington, County of Newcastle, DE (US).

(72) Inventor; and

(75) Inventor/Applicant (*for US only*): **GIESSEN, Charles, Richard** [US/GB]; 32 South Eaton Place, London SW1 9JJ (GB).

(74) Agent: **MURGITROYD & COMPANY**; 373 Scotland Street, Glasgow G5 8QA (GB).

(81) Designated States (*national*): AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CR, CU, CZ, DE, DK, DM, DZ, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, US, UZ, VN, YU, ZA, ZW.

(84) Designated States (*regional*): ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

Published:

— *Without international search report and to be republished upon receipt of that report.*

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

(54) Title: **SYSTEM FOR FACILITATING TRANSACTIONS ON AN EXCHANGE**

(57) Abstract: A broker-to-broker system enables originating brokers to send orders for execution to fulfilling brokers using broker-to-broker system secure networks and systems and for the resulting transactions to be cleared and settled in the name of the two parties using broker-to-broker system to manage the process. Locale-specific format conversions, such as for securities numbers, language, and currencies, are automatically performed.

WO 01/42949 A2

1 SYSTEM FOR FACILITATING TRANSACTIONS ON AN EXCHANGE

2

3 FIELD OF THE INVENTION

4 The present invention relates to a system which
5 may be used to facilitate transactions on an
6 exchange.

7

8 BACKGROUND OF THE INVENTION

9 There are many types of exchange on which
10 members communicate to transact business. For
11 example, members of financial exchanges may trade
12 money or an equivalent in exchange for an instrument
13 or commodity. Other types of trading exchanges
14 include the many electronic business-to-business
15 exchanges and business-to-consumer exchanges for
16 trading goods and services. Such business-to-
17 business exchanges typically adopt either a vertical
18 or a horizontal structure. Vertical exchanges are
19 based on specific industry sectors, e.g. aerospace,
20 automotive or chemicals. An example of a vertical

1 business-to-business exchange is GMtradexchange.com
2 on which parts for the automotive industry are
3 traded. Horizontal business-to-business exchanges
4 are organized around the products and services
5 provided which typically span more than a single
6 industry sector. An example of a horizontal
7 business-to-business exchange is MRO.COM which
8 provides a market place for materials, repair and
9 operations goods and services.

10 Exchanges of the above general types are used by
11 members. The term "member" used herein refers to a
12 party having access to an exchange in order to
13 transact business on it. The term "exchange" used
14 herein refers to any type of exchange on which
15 business can be transacted.

16 Although computer systems embodying the present
17 invention can be used with any type of exchange, the
18 exemplary embodiment described herein relates to
19 securities trading exchanges (including traditional
20 stock exchanges and alternative trading systems
21 (ATSS) and electronic crossing networks (ECNs)).
22 Principals or agents such as stock brokers,
23 investment banks or appropriately regulated asset
24 managers having direct or indirect access to
25 securities exchanges in order to buy and sell on
26 behalf of investors are referred to herein as
27 "brokers". For the purposes of the description the
28 term "broker" is synonymous with the term "Member".

29 In recent years, the U.S. market for equity
30 securities has experienced dramatic growth. This
31 growth is evidenced by the trading volumes in the

1 NASDAQ, NYSE and AMEX markets, as well as trading
2 volume in the so-called Third Market:

3 The average daily volume of securities traded in
4 the NASDAQ increased from 225.0 million shares in
5 December 1992 to 867.1 million shares in December
6 1999.

7 The average daily volume of securities traded on
8 the NYSE increased from 222.2 million shares in
9 December 1992 to 692.8 million shares in December
10 1998.

11 The average daily volume of securities traded on
12 the AMEX increased from 14.2 million shares in
13 December 1992 to 40.21 million shares in December
14 1998.

15 The trading volume and growth in all of these
16 markets are driven by many factors, including
17 increased cash flows into equity-based mutual funds;
18 historic high returns in U.S. equity markets; the
19 emergence and rapid growth of on-line discount
20 brokers; technological innovations, such as the
21 emergence of the Internet, and reduced transaction
22 costs.

23 While large trading volumes have provided
24 brokers with an opportunity to spread fixed costs
25 over a larger number of trades, net profits per
26 trade, however, have declined. As a result,
27 broker-dealers and other financial institutions are
28 seeking new trading methodologies to identify and
29 take advantage of the profit opportunities
30 represented by each trade. These broker-dealers also
31 seek to increase their order flow. Order flow is the

1 volume of buy and sell orders received by a
2 broker-dealer. Increased order flow provides market
3 makers with a greater number of trades and increased
4 trading profit opportunities. To succeed, these
5 broker-dealers require sophisticated trading
6 methodologies, computerized trading systems, and risk
7 management practices that can handle increased order
8 flow while maintaining low costs per trade. They also
9 need personnel with the requisite expertise to
10 deliver superior trade executions and customer
11 service.

12 For example, people who need to buy/sell
13 securities for investment purposes, i.e., Investors,
14 issue their orders to buy/sell securities to a
15 financial advisor/intermediary who is authorized to
16 carry out securities dealings, e.g. a broker or bank.
17 The broker then fulfils the Investor's order
18 generally in one of two ways:

19 (a) If the broker has access to an exchange
20 where the securities are transacted, the fulfilling
21 broker goes into that market and executes the order.
22 This action is the Domestic activity. The broker then
23 delivers the proceeds to the investor. This is the
24 "traditional" stock-brokering activity, which has
25 been taking place for the last 70-100 years;

26 (b) If the broker does not have access to an
27 exchange where the securities are transacted,
28 typically the case for small or medium-sized brokers
29 when asked to obtain foreign securities, the broker
30 communicates with another broker who is a member and
31 that broker executes the order. In this case the

1 first broker is a customer of the second broker. In
2 this case the second broker delivers the proceeds to
3 the first broker, who then in turn delivers the
4 proceeds to the investor. However, logistical
5 barriers currently make cross-border securities
6 transactions difficult, error-prone and expensive.

7

8 SUMMARY OF THE INVENTION

9 Systems embodying the present invention seek to
10 provide improved systems for facilitating
11 transactions.

12 According to a first aspect of the present
13 invention there is provided a method of operating a
14 computer system to facilitate transactions on an
15 exchange, a transaction being performed by a
16 fulfilling member having access to the exchange on
17 behalf of an originating party not having access to
18 the exchange, the method comprising:

19 receiving at a first interface of a processing
20 system at least one information item of an electronic
21 transaction proposal from an originating party;

22 transmitting at least one information item
23 relating to the electronic transaction proposal from
24 a second interface of the processing system to a
25 fulfilling member;

26 generating at the processing system settlement
27 criteria to be accepted by the originating party and
28 the fulfilling member; and

29 receiving from each of the originating party and
30 the fulfilling member an indication of acceptance of

1 the settlement criteria generated by the processing
2 system.

3 Preferably the step of receiving the at least
4 one information item of the electronic transaction
5 proposal comprises receiving an information item
6 indicating what is to be transacted. The step of
7 receiving the at least one information item of the
8 electronic transaction proposal may also comprise
9 receiving an information item identifying a
10 designated fulfilling member to perform the
11 transaction. The settlement criteria generated by
12 the processing system may be based on settlement
13 information received from the originating party and
14 the fulfilling member. The settlement criteria
15 generated by the processing system may also be based
16 on stored settlement information accessible by the
17 processing system. The settlement information
18 received from each of the originating party and the
19 fulfilling member includes an indication of
20 acceptance of the settlement criteria generated by
21 the processing system. The processing system can
22 generate settlement instructions to be issued on
23 behalf of the originating party and the fulfilling
24 member.

25 Preferably settlement instructions are generated
26 responsive to said indications of acceptance of the
27 settlement criteria.

28 In preferred embodiments a fulfilling member can
29 request a modification to at least part of an
30 electronic transaction proposal from an originating
31 party.

1 The step of receiving the at least one
2 information item of the electronic transaction
3 proposal may comprise receiving information items
4 relating to a proposed transaction selected from one
5 or more of the following:

6 a transaction type indicator;
7 a quantity indicator;
8 a price condition; and
9 timing information.

10 The timing information can indicate a proposed
11 transaction date and/or a proposed settlement date.

12 The step of receiving the at least one
13 information item of the electronic transaction
14 proposal may comprise receiving information items
15 relating to the proposed transaction from the
16 originating member in two or more sub-steps. For
17 example, an originating party may provide a first
18 information item in a first sub-step and the
19 processor system can supply a further relevant
20 information item responsive thereto. The processing
21 system can supply a plurality of further relevant
22 information items from which the originating party
23 can select. In another case, an originating member
24 provides a first information item indicating what is
25 to be transacted in a first sub-step and the
26 processing system prompts the originating party to
27 select from further information items representing
28 exchange options in a further sub-step.

29 The processing system can convert at least a
30 portion of an electronic transaction proposal from a
31 first format appropriate to the originating party

1 into a second format appropriate to the fulfilling
2 member. For example, the processing system converts
3 a transactional term in a first language appropriate
4 to the originating party into a second language
5 appropriate to the fulfilling member. The processing
6 system may convert a price indication in a first
7 currency appropriate to the originating party into a
8 corresponding price indication in a second currency
9 appropriate to the fulfilling member. If the
10 exchange is a securities exchange the processing
11 system can convert a first security identifier
12 appropriate to the originating party into a second
13 security identifier appropriate to the fulfilling
14 member. The first format is selected by the
15 originating party. The second format is selected by
16 the fulfilling member. The processing system can
17 also transfer an information item between the first
18 interface and the second interface without altering
19 the format of the information item.

20 Advantageously, preferred embodiments of the
21 processing system can generate a predetermined signal
22 to alert a member of a change in status of a proposed
23 transaction.

24 In many embodiments, the fulfilling member also
25 functions as an originating party by submitting
26 transaction proposals for execution on a further
27 exchange. Likewise, the originating member also
28 functions as a fulfilling member by performing a
29 transaction on a further exchange.

30 A computer system for facilitating transactions
31 on an exchange, a transaction being performed by a

1 fulfilling member having access to the exchange on
2 behalf of an originating party not having access to
3 the exchange, the system comprising:

4 a first processing system interface adapted to
5 receive one or more information items of an
6 electronic transaction proposal from an originating
7 party;

8 a second processing system interface adapted to
9 transmit one or more information items relating to
10 the electronic transaction proposal to a fulfilling
11 member;

12 a processing system for routing communications
13 including information items to and from the first and
14 second interfaces, the processing system being
15 operable to generate settlement criteria to be
16 accepted by the originating party and the fulfilling
17 member, and wherein the processing system is arranged
18 to receive first and second information items
19 indicating acceptance of the settlement criteria by
20 each of the originating party and the fulfilling
21 member.

22 According to another aspect of the present
23 invention there is provided a computer system for
24 facilitating transactions on an exchange, a
25 transaction being performed by a fulfilling member
26 having access to the exchange on behalf of an
27 originating party not having access to the exchange,
28 the system comprising:

29 a first processing system interface adapted to
30 receive one or more information items of an

1 electronic transaction proposal from an originating
2 party;

3 a second processing system interface adapted to
4 transmit one or more information items relating to
5 the electronic transaction proposal to a fulfilling
6 member;

7 at least one further processing system interface
8 adapted to issue settlement instructions in respect
9 of a transaction performed by the fulfilling member;
10 and

11 a processing system for routing communications
12 including information items to and from the first and
13 second interfaces, the processing system being
14 operable to generate settlement instructions to be
15 issued via the at least one further interface on
16 behalf of the originating member and the fulfilling
17 member.

18 According to another aspect of the present
19 invention there is provided a method of operating a
20 computer system to facilitate transactions on an
21 exchange, a transaction being performed by a
22 fulfilling member having access to the exchange on
23 behalf of an originating party not having access to
24 the exchange, the method comprising:

25 receiving at a first interface of a processing
26 system an information item of an electronic
27 transaction proposal from an originating party, said
28 information item being in a first form appropriate
29 for the originating party;

30 converting said information item of the
31 electronic transaction proposal from the first form

1 into a second form appropriate to a fulfilling
2 member; and

3 transmitting the information item in the second
4 form from a second interface of the processing system
5 to a fulfilling member.

6 According to another aspect of the present
7 invention there is provided a computer system for
8 facilitating transactions on an exchange, a
9 transaction being performed by a fulfilling member
10 having access to the exchange on behalf of an
11 originating party not having access to the exchange,
12 the system comprising:

13 a first processing system interface adapted to
14 receive an information item of an electronic
15 transaction proposal from an originating party, said
16 information item being in a first form appropriate to
17 the originating party;

18 a processing system operable to convert the
19 information item of the electronic transaction
20 proposal from the first form into a second form
21 appropriate to a fulfilling member; and

22 a second processing system interface adapted to
23 transmit the information item in the second form to
24 the fulfilling member.

25 According to another aspect of the present
26 invention there is provided a computer readable
27 medium having stored therein a set of general purpose
28 routines for facilitating transactions on exchanges,
29 the computer readable medium comprising:

1 a first routine for receiving at least one
2 information item of a transaction proposal from an
3 originating party;

4 a second routine for transmitting at least one
5 information item of the transaction proposal to a
6 fulfilling member;

7 a third routine for generating first and second
8 settlement criteria to be accepted by said
9 originating party and said fulfilling member; and

10 a further routine for receiving first and second
11 indications of acceptance of said settlement criteria
12 from said originating party and said fulfilling
13 member.

14 According to another aspect of the present
15 invention there is provided computer program code for
16 facilitating transactions on an exchange, a
17 transaction being performed by a fulfilling member
18 having access to an exchange on behalf of an
19 originating party not having access to the exchange,
20 the program code comprising:

21 a first set of instructions for receiving at
22 least one information item of a transaction proposal
23 from an originating party;

24 a second set of instructions for transmitting at
25 least one information item of the transaction
26 proposal to a fulfilling member;

27 a third set of instructions for generating
28 settlement criteria to be agreed by each of said
29 originating party and said fulfilling member and for
30 receiving first and second indications of agreement

1 from said originating party and said fulfilling
2 member.

3 According to another aspect of the present
4 invention, there is provided a method of facilitating
5 transactions on an exchange, a transaction being
6 performed by a fulfilling member having access to the
7 exchange on behalf of an originating party not having
8 access to the exchange, the method comprising:

9 receiving at least one information item of a
10 transaction proposal from an originating party;
11 transmitting at least one information item of
12 the transaction proposal to a fulfilling member;
13 generating settlement criteria to be accepted by
14 the originating party and the fulfilling member; and
15 receiving indications of acceptance of the
16 settlement criteria from each of said originating
17 party and said fulfilling member.

18 Preferably, the method includes generating first
19 and second settlement instructions on behalf of said
20 originating party and said fulfilling member
21 responsive to receipt of said indications of
22 acceptance.

23 According to another aspect of the present
24 invention there is provided a system for facilitating
25 transactions on an exchange, a transaction being
26 performed by a fulfilling member having access to the
27 exchange on behalf of an originating party not having
28 access to the exchange, the system comprising:

29 means for receiving a transaction proposal from
30 an originating party;

1 means for transmitting a transaction proposal to
2 a fulfilling member;

3 means for receiving accepted settlement
4 particulars from said originating party and said
5 fulfilling member;

6 means for generating first and second settlement
7 instructions on behalf of said originating member and
8 said fulfilling member responsive to receipt of said
9 accepted settlement particulars; and

10 means for issuing said first and second
11 settlement instructions.

12 According to another aspect of the present
13 invention there is provided a method of facilitating
14 transactions on an exchange, a transaction being
15 performed by a fulfilling member having access to the
16 exchange on behalf of an originating party not having
17 access to the exchange, the method comprising:

18 receiving an information item of a transaction
19 proposal from an originating party, said information
20 item being in a first form appropriate for the
21 originating party;

22 converting said information item of the
23 transaction proposal from the first form into a
24 second form appropriate to a fulfilling member; and
25 transmitting the information item in the second
26 form to a fulfilling member.

27 According to another aspect of the present
28 invention there is provided a system for facilitating
29 transactions on an exchange, a transaction being
30 performed by a fulfilling member having access to the

1 exchange on behalf of an originating party not having
2 access to the exchange; the method comprising:

3 means for receiving an information item of a
4 transaction proposal from an originating party, said
5 information item being in a first form appropriate to
6 the originating party;

7 means for converting said information item of
8 the transaction proposal from the first form into a
9 second form appropriate to a fulfilling member; and

10 means for transmitting the information item in
11 the second form to a fulfilling member.

12 According to another aspect of the present
13 invention there is provided computer program code for
14 facilitating transactions on an exchange, a
15 transaction being performed by a fulfilling member
16 having access to an exchange on behalf of an
17 originating party not having access to the exchange,
18 the program code comprising:

19 a first set of instructions for receiving an
20 information item of a transaction proposal from an
21 originating party, said information item being in a
22 first form appropriate for the originating party;

23 a second set of instructions for converting said
24 information item of the transaction proposal from the
25 first form into a second form appropriate to a
26 fulfilling member; and

27 a third set of instructions for transferring the
28 information item in the second form to a fulfilling
29 member.

30 According to another aspect of the present
31 invention there is provided a computer readable

1 medium having stored therein a set of general purpose
2 routines for facilitating transactions on exchanges,
3 the computer readable medium comprising:
4 a first routine for receiving an information
5 item of a transaction proposal from an originating
6 party, said information item being in a first form
7 appropriate for the originating party;
8 a second routine for converting said information
9 item of the transaction proposal from the first form
10 into a second form appropriate to a fulfilling
11 member; and
12 a third routine for transmitting the information
13 item in the second form to a fulfilling member.
14 Preferred embodiments provide a system, method,
15 and software for broker-to-broker trading, in which
16 an order for a proposed transaction is received from
17 an originating member, indicating a designated
18 fulfilling broker. The order is in a format
19 appropriate to the originating broker. The order is
20 converted into in a format appropriate to the
21 fulfilling broker and transmitted to the fulfilling
22 broker and settlement instructions are issued from a
23 central location. As a result, preferred embodiments
24 deal with local-specific conversion and settlement
25 issues are automatically handled, thereby increasing
26 the efficiency of cross-border transactions, reducing
27 transaction costs, fostering increased transparency,
28 and increasing the volume and liquidity of
29 international securities transactions. Preferred
30 embodiments thus provide a high value, robust
31 messaging and transaction processing infrastructure

1 that will enable executing-brokers ("Fulfilling
2 Members" (FMs)) to service orders directed to them
3 and provide brokers placing orders ("Originating
4 Members" (OMs)) variable lower-cost access to
5 securities markets and transaction services
6 worldwide.

7 Still other objects and advantages of preferred
8 embodiments of the present invention will become
9 readily apparent from the following detailed
10 description, simply by way of illustration of the
11 best modes contemplated of carrying out the
12 invention. As will be realized, the invention is
13 capable of other and different embodiments, and its
14 several details are capable of modifications in
15 various obvious respects, all without departing from
16 the invention. Accordingly, the drawings and
17 description are to be regarded as illustrative in
18 nature, and not as restrictive.

19

20 BRIEF DESCRIPTION OF THE DRAWINGS

21 The present invention is illustrated by way of
22 example, and not by way of limitation, in the figures
23 of the accompanying drawings and in which like
24 reference numerals refer to similar elements and in
25 which:

26 FIGURE 1 is a schematic diagram illustrating a
27 broker-to-broker communication system embodying the
28 present invention;

29 FIGURE 2 is a schematic diagram illustrating key
30 information flows in a system according to Figure 1;

1 FIGURE 3 is a schematic diagram of an end
2 terminal for use in the broker-to-broker system of
3 Figure 1;

4 FIGURE 4 is a schematic diagram of a computer
5 system implementing the broker-to-broker system of
6 Figure 1;

7 FIGURE 5 is a flow chart illustrating connection
8 and display update processes of the computer system
9 of Figure 4;

10 FIGURE 6 is a flow chart illustrating order
11 placement processes of the computer system of Figure
12 4;

13 FIGURE 7 is a flow chart illustrating order
14 acceptance/rejection processes of the computer system
15 of Figure 4;

16 FIGURE 8 is a flow chart illustrating execution
17 reporting processes of the computer system of Figure
18 4;

19 FIGURE 9 is a flow chart illustrating settlement
20 initiation and processing procedures of the computer
21 system of Figure 4;

22 FIGURE 10 is a schematic diagram illustrating
23 another example of a computer system suitable for
24 implementing the broker-to-broker communication
25 system of Figure 1;

26 FIGURE 11 is a schematic diagram of an end
27 terminal and local network for use in the computer
28 system of Figure 10;

29 FIGURE 12 schematically illustrates a first
30 interface architecture of the computer system of
31 Figure 10;

1 FIGURE 13 schematically illustrates a second
2 interface architecture of the computer system of
3 Figure 10;

4

5 DESCRIPTION OF THE PREFERRED EMBODIMENT

6 A broker-to-broker trading system and
7 methodology are described. In the following
8 description, for the purposes of explanation,
9 numerous specific details are set forth in order to
10 provide a thorough understanding of the present
11 invention. It will be apparent, however, to one
12 skilled in the art that the present invention may be
13 practiced without these specific details. In other
14 instances, well-known structures and devices are
15 shown in block diagram form in order to avoid
16 unnecessarily obscuring the present invention.

17

18 DEFINITION OF TERMS

19 The financial industry is beset with terminology
20 that varies with jurisdiction. Thus, in order to
21 prevent confusion and because the broker-to-broker
22 system is an entirely novel arrangement within the
23 financial industry, i.e. it is not a bank, brokerage,
24 exchange, advisor, or any other kind of conventional
25 financial related entity, it is vital that clarity of
26 purpose, role and responsibility of the components,
27 systems and entities within the broker-to-broker
28 system be maintained at all times.

29 **Originating Party/Member** The originating
30 party/member is the entity on the broker-to-broker
31 system's network that originates orders into the

1 network. The originating party may be any
2 institution appropriately regulated to place orders
3 with a fulfilling member on behalf of themselves or
4 third parties. The originating member is called thus
5 because this entity can be a member of an exchange
6 who starts the process off by entering an order into
7 the system, i.e. it originates the order. It is
8 assumed that typically orders entered into the
9 broker-to-broker system network have been entered
10 because the originating member has been requested by
11 a customer to purchase/sell securities. Strictly
12 speaking however, there is no requirement for the
13 originating member to enter only their customer's
14 requests, in addition some members may conduct
15 extensive "own-account"/proprietary trading using the
16 broker-to-broker system's network to send orders for
17 execution. In examples in this document the
18 originating party/member is designated as Party A.

19 **Fulfilling Member** The fulfilling member is
20 the entity on the broker-to-broker system's network
21 which receives orders from the originating member and
22 then executes against those orders, i.e. the order is
23 the desire to do something and fulfillment is the
24 satisfying of that desire. In most securities
25 markets, the fulfilling member has the following
26 options available to them as to how to fulfil the
27 order or parts thereof.

28 - On Exchange, i.e. by buying/selling securities
29 from/to another exchange member. The fulfilling
30 member may or may not expose the order to the entire
31 market depending on the rules of the market.

1 - Off Inventory, i.e. by buying/selling
2 securities from/to the fulfilling member own
3 inventory of securities.

4 - Agency Cross, i.e. when the fulfilling member
5 determines that there are two or more orders on
6 opposite sides of each other, (i.e. one order is for
7 a sale and another order is for a purchase of the
8 same set of securities), the member matches the two
9 orders.

10 - Third party crossing network, i.e. when the
11 fulfilling broker sends the order into a third party
12 crossing network such as Instinet, and Posit.

13 In examples in this document the fulfilling
14 member is designated as Party B.

15 **Member** Members are customers (users) of the
16 broker-to-broker system having access to one exchange
17 or another and are usually brokers acting as
18 transacting agents for investors. When referring to
19 the clients of the broker-to-broker system's member's
20 the term Investor will be used implying the eventual
21 beneficiary of the proceeds of transactions.
22 Generally, investors have no contractual or any other
23 kind of relationship with the broker-to-broker
24 system; an investor's relationship legal or otherwise
25 exists between the investor and their financial
26 advisor(s).

27 **Investor** The term used for entities that have
28 contracted with one of the broker-to-broker system's
29 members to carry out services. Investors do not
30 usually contract with the broker-to-broker system,
31 nor would they directly use the services of the

1 broker-to-broker system - see "Product" below for how
2 investors may use technology created by the broker-
3 to-broker system, in order to access the broker-to-
4 broker system's services/products even indirectly,
5 the investor has a relationship with the broker-to-
6 broker system customer who is responsible or liable
7 for all the actions of the investor.

8 **Street** The term used to denote entities
9 within the financial community that will interact
10 with the broker-to-broker system's members typically
11 to provide them with research and execution
12 counterparties

13 **Service** The term used to denote ongoing
14 activities carried out by the broker-to-broker system
15 personnel, and/or its agents and partners for which
16 the broker-to-broker system's member's will pay some
17 kind of fee.

18 **Product** The term used to denote an item(s)
19 which may be sold to, licensed to, leased to, or
20 rented to the broker-to-broker member which that
21 member may then use to offer a service to his/her
22 clients, e.g. investors. Optionally the broker-to-
23 broker system through one of its facilities
24 management operating subsidiaries and in conjunction
25 with its partners and agents may be selected by the
26 member to install, configure and operate the product
27 on behalf of and in the name of that member.

28 **Order** An order is an instruction from one
29 party to another party to obtain/dispose of
30 securities on behalf of the issuing party in return
31 for some payment. Typically unless an order is

1 fulfilled, no transaction will be deemed to have
2 taken place.

3 **Proceeds** Proceeds are the cash monies or
4 securities which parties to a transaction will
5 deliver/receive at the conclusion of the transaction.

6 **Settlement** Settlement is the process of
7 delivering the proceeds of a transaction into the
8 custody of the beneficiaries of the transaction, e.g.
9 cash into a bank account, securities into a
10 securities account.

11 **Execution** Whenever a securities transaction
12 takes place, an execution is deemed to have taken
13 place, i.e. one party agrees to sell securities to
14 another party in exchange for remuneration. The terms
15 "fill" or "completion" are also often used instead of
16 execution particularly when many small securities
17 transactions are required to fulfil an order.

18 **Custodian** Securities are not always held by
19 investors in the form of paper certificates. Most
20 developed markets insist that securities are held in
21 de-materialized form in a depository, (often a
22 computer system managed by some quasi-government
23 institution or association of banks, which keeps
24 track of who is the beneficial owner of the
25 security). In such cases, the depository may
26 nominate, approve and supervise certain commercial
27 organizations to provide the interface between the
28 depository and the owners of securities - these
29 organizations are the custodians who maintain custody
30 of the securities on behalf of the actual owners.

1 (Custodians may also maintain vaults where paper
2 certificates are stored).

3 SYSTEM OVERVIEW

4 FIG. 1 represents a schematic overview of a
5 preferred communications system. The broker-to-
6 broker computer system is an order routing and
7 settlement facilitation system, which comprises an
8 integrated network of desktop applications through
9 which an originating member 204 seeking fulfillment
10 of an investor 210's securities transaction order may
11 enter the order for delivery to a designated
12 fulfilling member 208 who is qualified to fulfill the
13 order or part thereof on the basis of prevailing
14 conditions in the fulfilling members' market 212,
15 subject to the conditions specified by the
16 originating member 204. In this embodiment, a
17 broker-to-broker network arrangement comprises two
18 secure interfaces OMIF 202 and FMIF 206 connecting
19 originating members 204 and fulfilling members 208,
20 respectively, to a broker-to-broker processing system
21 200 (having "order management" capabilities), which
22 processing system acts as a transaction manager
23 tracking the status of orders. Additional integrated
24 applications generate settlement instructions,
25 collect and report status information from other
26 parts of the broker-to-broker system, provide access
27 for the correction of problems of the broker-to-
28 broker system, and perform data storage and retrieval
29 functions. Members are able to use the broker-to-

1 broker system as originating members 204 or
2 fulfilling members 208, or both.

3 It will be appreciated that many originating
4 members 204 and fulfilling members 208 can be
5 connected to the broker-to-broker processing system
6 200 and that the interfaces 202 and 206 can be any
7 type necessary to achieve this.

8 Using the interfaces 202 and 206 connecting them
9 to the broker-to-broker system, originating members
10 204 may enter information concerning a proposed
11 transaction. If all required information is properly
12 provided, the order is then sent to an order
13 management system for routing and delivery to the
14 fulfilling member 208. As this embodiment is a
15 broker-to-broker system designed to enhance the
16 ability of domestic originating members 204 to send
17 appropriate orders to fulfilling members 208
18 qualified to execute such trades on foreign markets,
19 the processing System 200 is able to perform certain
20 data conversion or translation services of a clerical
21 or ministerial nature when a cross-border or
22 inter-market trade calls for such services.

23 The combined order routing and settlement
24 management features of the broker-to-broker
25 processing system 200 afford a seamless and efficient
26 inter-exchange order management system. Conversion
27 service features of the processing system 200 provide
28 opportunities for increased efficiency of
29 cross-border and inter-market transactions and thus
30 contribute to broader access to securities markets
31 worldwide, in particular because the processing

1 system 200 offers those capabilities at lower
2 variable costs to members who have previously been
3 less able to fully take advantage of advanced
4 technologies to overcome the costs and other burdens
5 arising from the numerous additional variables
6 associated with international securities
7 transactions. It is particularly advantageous that
8 settlement instructions are issued from a single
9 source. Accordingly, the use of the broker-to-broker
10 network by members may further increase the
11 transparency of international securities markets and
12 afford to investors through their brokers increased
13 liquidity and efficiency in conducting securities
14 transactions.

15 All communications are routed through the
16 broker-to-broker processing system 200. All
17 communications sent to and from the broker-to-broker
18 system may be encrypted using 128 bit SSL technology
19 to ensure complete confidentiality. Once an order is
20 received, a fulfilling member 208 designated by the
21 originating member 204 will review the information
22 for completeness and accuracy, as will be described
23 in more detail hereinafter. The fulfilling member 208
24 can then determine in its sole discretion whether to
25 accept or reject the order. Once an order is
26 accepted, the fulfilling member 208 will then seek to
27 execute the order on the specified market satisfying
28 the terms of the order. The fulfilling member and the
29 originating member communicate through the broker-to-
30 broker system or through any other means they may
31 choose as to the status of fulfillment of the order

1 and any modifications thereto. When the members agree
2 that the order has been fulfilled (or the unfulfilled
3 portion cancelled), information concerning the
4 transaction is sent to the originating member 204 for
5 review. The agreed transaction can then be submitted
6 by both members to the settlement management portion
7 of the broker-to-broker processing system 200 which
8 generates settlement instructions, based entirely on
9 transaction details and settlement account (and
10 preference) information previously provided by
11 members, for transmission to settlement agents
12 designated exclusively by the respective members.

13 Additional functions to be performed by the
14 broker-to-broker processing system 200 include the
15 generation of execution reports, and similar
16 functions as may be desired by brokers using the
17 broker-to-broker computer system.

18

19 KEY SYSTEM FLOWS

20 FIG. 2 schematically depicts the system flows
21 for the broker-to-broker system, showing how an order
22 moves through the various network elements until the
23 proceeds are delivered to one of the broker-to-broker
24 system's members.

25 The investor issues an order to his/her
26 financial originating member who enters the order
27 into an originating member interface OMIF 202.
28 Referring to Fig. 2, the order 10 is then transmitted
29 through the processing system 200 via an order
30 management sub-system 16 to a fulfilling member via a
31 fulfilling member interface FMIF 206. The fulfilling

1 member executes the order in his/her market and
2 notifies the originating member of the results by
3 sending a fill message 12 to the originating member
4 via the originating member interface OMIF 202. When
5 both members agree that the order has been fulfilled,
6 a message 14 detailing the transaction is sent to a
7 settlement management system 20 which completes the
8 process by issuing settlement instructions 22
9 according to the SWIFT protocol, or another suitable
10 protocol, thereby ensuring that both parties
11 deliver/receive the securities, and that monies are
12 paid by the members to/from each other.

13 At all stages in the transaction cycle, a
14 customer service system 24, is notified by all of the
15 other network elements of the processing system 200
16 of the status of the transaction so that inquiries
17 may be easily and rapidly dealt with.

18

19 HARDWARE OVERVIEW: END TERMINAL

20 FIG. 3 is a block diagram illustrating a
21 computer system 400/402 which is used as an end
22 terminal of both the originating member and the
23 fulfilling member in the exemplary embodiment
24 described herein. Computer system 400/402 includes a
25 bus 102 or other communication mechanism for
26 communicating information, and a processor 104
27 coupled with bus 102 for processing information.
28 Computer system 400/402 also includes a main memory
29 106, such as a random access memory (RAM) or other
30 dynamic storage device, coupled to bus 102 for
31 storing information and instructions to be executed

1 by processor 104. Main memory 106 also may be used
2 for storing temporary variables or other intermediate
3 information during execution of instructions to be
4 executed by processor 104. Computer system 400/402
5 further includes a read only memory (ROM) 108 or
6 other static storage device coupled to bus 102 for
7 storing static information and instructions for
8 processor 104. A storage device 110, such as a
9 magnetic disk or optical disk, is provided and
10 coupled to bus 102 for storing information and
11 instructions.

12 Computer system 400/402 may be coupled via bus
13 102 to a display 112, such as a cathode ray tube
14 (CRT), for displaying information to a computer user.
15 An input device 114, including alphanumeric and other
16 keys, is coupled to bus 102 for communicating
17 information and command selections to processor 104.
18 Another type of user input device is cursor control
19 116, such as a mouse, a trackball, or cursor
20 direction keys for communicating direction
21 information and command selections to processor 104
22 and for controlling cursor movement on display 112.
23 This input device typically has two degrees of
24 freedom in two axes, a first axis (e.g., x) and a
25 second axis (e.g., y), that allow the device to
26 specify positions in a plane.

27 In this embodiment communication with the
28 broker-to-broker processing system is facilitated by
29 browser software running on the computer system
30 400/402 by means of the processor 104 executing one
31 or more sequences of one or more instructions

1 contained in the main memory 106. Such instructions
2 may be read into main memory 106 from another
3 computer-readable medium, such as storage device 110.
4 Execution of the sequences of instructions contained
5 in main memory 106 causes processor 104 to perform
6 the process steps described herein. One or more
7 processors in a multi-processing arrangement may also
8 be employed to execute the sequences of instructions
9 contained in main memory 106.

10 The term "computer-readable medium" as used in
11 this context refers to any medium that participates
12 in providing instructions to processor 104 for
13 execution. Such a medium may take many forms,
14 including but not limited to, non-volatile media,
15 volatile media, and transmission media. Non-volatile
16 media include, for example, optical or magnetic
17 disks, such as storage device 110. Volatile media
18 include dynamic memory, such as main memory 106.
19 Transmission media include, for example, coaxial
20 cables, copper wire, fiber optics and radio
21 interfaces. Transmission media can thus also take
22 the form of acoustic or light waves, such as those
23 generated during radio frequency (RF) and infrared
24 (IR) data communications. Common forms of computer-
25 readable media include, for example, a floppy disk, a
26 flexible disk, hard disk, magnetic tape, any other
27 magnetic medium, a CD-ROM, DVD, any other optical
28 medium, punch cards, paper tape, any other physical
29 medium with patterns of holes, a RAM, a PROM, and
30 EPROM, a FLASH-EPROM, any other memory chip or

1 cartridge, a carrier wave as described hereinafter,
2 or any other medium from which a computer can read.

3 Various forms of computer readable media may be
4 involved in carrying one or more sequences of one or
5 more instructions to processor 104 for execution.
6 For example, the instructions may initially be borne
7 on a magnetic disk of a remote computer. The remote
8 computer can load the instructions into its dynamic
9 memory and send the instructions over a communication
10 link such as telecommunications link with or without
11 using a modem. A modem local to computer system
12 400/402 can receive the data on a communication link
13 and use an infrared transmitter to convert the data
14 to an infrared signal. An infrared detector coupled
15 to bus 102 can receive the data carried in the
16 infrared signal and place the data on bus 102. Bus
17 102 carries the data to main memory 106, from which
18 processor 104 retrieves and executes the
19 instructions. The instructions received by main
20 memory 106 may optionally be stored on storage device
21 110 either before or after execution by processor
22 104.

23 Computer system 400/402 also includes a
24 communication interface 118 coupled to bus 102. For
25 example, communication interface 118 may be an
26 integrated services digital network (ISDN) card or a
27 modem to provide a data communication connection to a
28 telecommunications line. Communication interface 118
29 provides a two-way data communication coupling to a
30 communication line 120 comprising part of the
31 originating member interface OMIF 202 or the

1 fulfilling member interface FMIF 206, depending on
2 whether the computer 400/402 is an end terminal
3 belonging to an originating member or a fulfilling
4 member. Fixed or leased telecommunications lines are
5 used to connect the end terminals 400/402 to the
6 broker-to-broker processing system 200. The minimum
7 fixed-line speed available for the broker-to-broker
8 system is at present 64Kbits/second. Higher speeds
9 are installed in proportion to the number of
10 concurrent broker-to-broker system sessions in use at
11 the member's location, and the response times which
12 they wish to achieve. As a rule of thumb, 2 sessions
13 can be concurrently handled per 64Kbit/sec segment.

14 The interfaces OMIF 202 and FMIF 206 thus
15 connect originating and fulfilling members,
16 respectively, to the broker-to-broker processing
17 system 200 which acts as an order placing/accepting
18 mechanism, a transaction manager for tracking the
19 status of orders, and a means of automatically
20 issuing settlement instructions. Additional
21 integrated applications collect and report status
22 information from other parts of the system, provide
23 access for maintenance and the correction of system
24 problems, and perform data storage and retrieval
25 functions. End terminals 400 of the originating and
26 fulfilling member interfaces OMIF 202 comprise a web
27 technologies based front-end (or graphical user
28 interface) which the members interact with and which
29 connects to an application core for processing the
30 data entered by the respective member. Preferred
31 embodiments also include a computer-to-computer

1 applications programmer interface OMIF-API so the
2 originating member can interface his/her computer
3 systems directly into the broker-to-broker system.

4 FMIF 206, the system which interfaces the
5 broker-to-broker system to the fulfilling broker 208,
6 provides him/her with the ability to interact with
7 the broker-to-broker system and its other customers.
8 Like OMIF 202, FMIF 206 comprises: a web technologies
9 based front-end, through which the fulfilling member
10 interacts with the broker-to-broker system and an
11 application core. Preferred embodiments also include
12 a computer-to-computer applications programmer
13 interface, FMIF-API, which the fulfilling member may
14 use to interface his/her systems directly into the
15 broker-to-broker system network so as to eliminate
16 re-keying of data.

17 In this embodiment, the front-end software (GUI)
18 uses the following standards: Netscape Navigator 4.5,
19 Microsoft Internet Explorer 4; HTML 4: JavaScript;
20 JAVA; 128 Bit SSL. The application core may be built
21 using the following standards: Netscape Commerce or
22 Apache Server; CGI with Perl and C/C++ 504; Java
23 servlets; Java servlet pages; application server
24 (e.g. WebLogic); SUN Solaris or Linux Operating
25 System; Oracle Database; Tellurian Secure Sockets.

26 The target technical environment at the
27 originating/fulfilling member end terminal will thus
28 be Netscape Navigator 4.5 or Microsoft Internet
29 Explorer 4.01, higher versions may be used. Cookies,
30 small files stored on the user's local hard-drive,
31 are commonly used by web based applications to store

1 data about the user and their preferences so that
2 when the user next logs on, the system can skip the
3 steps required to obtain that data for the new
4 session. The use of cookies within the OMIF 202 and
5 FMIF 206 interfaces can be severely restricted if not
6 eliminated so as to minimize any security risk. That
7 is, items like colors, and language preference, frame
8 sizes and positions can be safely stored in cookies,
9 but items such as security identifiers, default
10 fulfilling members and default execution venue are
11 preferably not be stored since knowledge of these
12 could be used to interpret an originating member's
13 transaction history.

14 The OMIF 202 and FMIF 206 are constructed so as
15 to require only minimal resources at the end
16 terminals 400,402. That is, they have an operating
17 "footprint" on the user's machine which is as
18 lightweight or small as possible. It must not be
19 considered the norm that users will be technically
20 literate and that they will have high powered
21 machines. A typical end terminal might be a Pentium
22 personal computer running the Microsoft Windows NT
23 operating system.

24

25 HARDWARE OVERVIEW: BROKER-TO-BROKER PROCESSING SYSTEM

26 Figure 4 illustrates the preferred architecture
27 for the broker-to-broker processing system 200.
28 Figure 4 also shows how the end terminals of an
29 originating member 400 and a fulfilling member 402
30 are connected to the broker-to-broker processing
31 system 200 by a secure network 404. The secure

1 network 404 comprises at least the communications
2 lines 120 from the end terminals 400 and 404.

3 The broker-to-broker processing system 200
4 comprises a web server 410, an application server 420
5 provided with a front end database 425, an order
6 management system (OMS) server 430 provided with an
7 OMS database 435, and a settlement management system
8 (SMS) server 440 provided with a SMS database 445.
9 The OMS database 435 and the SMS database 445 are
10 connected by a database link 452. The web server 410
11 is connected to a messaging server 480 and the
12 application server 420 is connected to a security
13 server 470. The interface between the web server 410
14 and the application server 420 operates based on the
15 Java RMI protocol. The order management system
16 server 430 is connected to the application server 420
17 via an interface defined according to the Orbix CORBA
18 protocol.

19 The web server 410 includes a submission
20 processor 412 and a page generator application 414.
21 The web server 410 supports a range of internet-based
22 server technologies including, for example, hyper
23 text mark-up language (HTML), JavaScript, extensible
24 mark-up language (XML), and other protocols for the
25 definition of page content, format and functionality.
26 It should be appreciated that the web server is not
27 connected to the internet in this embodiment.

28 The application server 420 includes a processor
29 (not shown), order and query data access software
30 422, order manager software 424 and order storage
31 software 423. The application server 420 operates

1 hosting and data access services for the web server
2 410 to create pages which are implemented as
3 Enterprise Java Beans. The front end database 425
4 connected to the application server 420 holds records
5 of orders 426, executions 427, requests 428 and user
6 control information 429. The application server 420
7 controls page generation through the web server 410
8 using the order manager software 420.

9 The order management system server 430 includes
10 a request processor 432, order publisher software 434
11 and an order processor 436. The OMS database 435
12 holds records of orders 437, executions 438 and
13 requests 439. The order management system server 430
14 can transfer data to the front end database 425 via
15 the order publishing software 434 and the order
16 storage software 423. Accordingly, data held in the
17 OMS database 435 can be replicated into the front end
18 database 425 following an access to the OMS database
19 435 by the order processor 436. The order management
20 system server 430 can also access the SMS database
21 445 via the database link 452.

22 The FIX server 490 is provided in order to
23 implement application programmer interfaces, as will
24 be explained in more detail hereinafter.

25 The settlement management system server 440 has
26 a trade processor 442 capable of generating
27 settlement instructions in the form of, for example,
28 SWIFT format messages 443 and fax messages 444. The
29 SMS database 445 holds records of settled trades 446,
30 stock information 447, market information 448, member
31 information 449 and commission rate information 450.

1 The settlement management system server 440 polls the
2 SMS database 445 at predetermined time intervals to
3 identify transactions for which settlement
4 instructions can be generated. The SMS database 445
5 receives an intra-day price feed every half-past the
6 hour, where the price given is the price of the hour
7 (e.g. the price at 10.00am is delivered at 10.30am).
8 The days closing prices are also fed in nightly.

9 As will be explained hereinafter, the settlement
10 management system server 440 need only become
11 involved when the settlement particulars have been
12 agreed by the parties to the transaction. Although
13 the various software and hardware components of the
14 broker-to-broker processing system 200 are
15 illustrated as separate functional components for
16 reasons of clarity, they may of course be provided in
17 a different configurations.

18 The application server 420 and the front end
19 database together define an "application subsystem"
20 of the computer system on which the core application
21 software runs. The core application software
22 provides content to the end terminals. The order
23 management system server 430 and the OMS database 435
24 together define an order management sub-system which,
25 as the name suggests, is responsible for validating
26 each originating member order; keeping track of the
27 status of orders; controlling, tracking, and applying
28 conversions/translations to the order; entering the
29 business rules associated with translations; and
30 dispatching completed orders for settlement
31 processing. Similarly, the settlement management

1 system server 440 and the SMS database 445 can be
2 regarded as comprising a settlement management sub-
3 system. The first action taken by the order
4 management sub-system is to apply stringent/rigorous
5 content validation to each order. The order
6 management sub-system can access information in the
7 settlement management sub-system, as may be required,
8 and provide messages and other content to the end
9 terminals via the application server 420, front end
10 database 425 and the web server 410.

11 Once an order passes the content validation
12 checks of the order management sub-system, it is
13 routed to the fulfilling member specified by the
14 originating member. All interactions with orders are
15 passed via the order management sub-system which
16 ensures that predefined rules of business are not
17 violated. For example, the order management sub-
18 system will prohibit members from entering an
19 executed quantity greater than the order quantity; it
20 will prohibit multiple members from applying changes
21 to the order simultaneously; it will ensure that the
22 state rules of an order cannot be violated.

23 The order management sub-system facilitates
24 and controls the following actions performed by
25 either the originating member (OM) and/or the
26 fulfilling member (FM) in respect of an order in the
27 state specified:

28
29
30
31

<u>State of Order:</u>	<u>OM</u>	<u>FM</u>
New Order Entered	Cancel	Accept; Reject
Accepted Order	Cancel; Modify	Fill; Cancel; Modify
OM (cancel) Request	Withdraw	Accept; Reject
FM (cancel) Request	Accept; Reject	Withdraw
OM Modification		
Request	Withdraw	Accept; Reject
FM Modification		
Request	Accept; Reject	Withdraw
OM Sign-off Request	Withdraw	Accept; Reject
FM Sign-off Request	Accept; Reject	Withdraw
FM (cancel) Fill		
Request	Accept; Reject	Withdraw
Cancelled	-	-
Rejected	-	-
Signed off	-	-

1

2 The order management sub-system also routes any
3 requests against an order from the originating party
4 to the receiving party; and conversely the
5 accept/rejection of a request is routed back to the
6 originating party, as will be explained further
7 hereinafter.

8

9

OPERATION

10 Because the front end software may be used at
11 many sites, and very often where the level of
12 technical support is minimal or self-help is the
13 norm, it preferably does not require the installation
14 of any components from disk/CDROM etc at the
15 customer's premises. As will be explained, the front

1 end software accessed via end terminals 400 of the
2 OMIF 202 may include the following frames or
3 capabilities: Order entry; order manager (indicating
4 status of orders, cancel, modify, sign-off); intra
5 broker-to-broker system member messaging (e-mail,
6 interactive text messaging, e.g. ICQ, IRC,
7 interactive voice, video messaging, e.g. net
8 meeting); transaction history report generation;
9 lookup facilities for securities information; look-up
10 facilities to locate other members on the broker-to-
11 broker system; and an interface to the broker-to-
12 broker system on-line electronic library.

13 The front end software accessed via the end
14 terminals 402 of the FMIF 406 includes the following
15 frames or capabilities: order acceptance; execution
16 entry; transaction manager (status of orders, cancel,
17 modify, cancel fill, sign off); intra broker-to-
18 broker system customer messaging; (e-mail;
19 interactive text messaging, e.g. ICQ, IRC;
20 interactive voice, video messaging, e.g. net
21 meeting;) transaction history report generation;
22 look-up facilities for securities information; Look-
23 up facilities to locate other members on the broker-
24 to-broker system; interface to the broker-to-broker
25 system on-line electronic library.

26 As the system is designed in part to enhance the
27 ability of domestic originating members to send
28 appropriate orders to fulfilling members qualified to
29 execute such trades on foreign markets, the system
30 can perform certain data conversion or translation
31 services of a clerical or ministerial nature when a

1 cross-border or inter-market trade calls for such
2 services. Accessed through drop-down menus viewed
3 on the GUI of the front end, the conversion service
4 options available to originating members include:
5 retrieval from data storage and inclusion in the
6 message of appropriate additional securities
7 identification numbers (e.g., the system can provide
8 the applicable ISIN number when given a particular
9 CUSIP number for securities identified for trading in
10 multiple jurisdictions); the reformatting of orders
11 to comply with the customs, dictates or market
12 vernacular of a particular exchange to be used by the
13 designated fulfilling member (e.g., "market held"
14 from a U.S. broker would be converted to "best," the
15 equivalent expression used by broker-dealers in Great
16 Britain); language translation (e.g., English to
17 French) of essential transaction terms (e.g., "buy"
18 or "sell"); and the generation and inclusion by the
19 system of additional available information requested
20 by an originating member that is not an element of a
21 proposed transaction, but offers convenience to the
22 parties, such as display of indicative-only currency
23 conversion data. The system also offers members an
24 option to include free form messages which are not
25 subject to any modification, translation or
26 conversion by the system.

27 Figure 5 is a flow chart illustrating how both
28 originating members and fulfilling members can view
29 relevant information (information specific to the
30 member) on their respective end terminals 400, 402.
31 The end terminals 400, 402, are in this embodiment

1 personal computers located at a site owned by the
2 member. As outlined above, the end terminals 400,
3 402, are provided with a browser application capable
4 of: displaying pages of order information; generating
5 messages; and generating reports. Originating and
6 fulfilling members establish a connection and log-on
7 to the broker-to-broker processing system 200 in the
8 same way. For security purposes every user of the
9 system use a physical security token unique to each
10 individual at the originating member's premises, such
11 as SecurID from RSA Dynamics, to identify themselves
12 to the system. This token and operating guide may be
13 delivered to the member using secure courier
14 facilities. Once in possession of the token, all that
15 may be required of the user may be to start up their
16 web browser and navigate to the broker-to-broker URL
17 . If there are software components to be installed
18 the broker-to-broker system will automatically detect
19 that condition and carry out the necessary operations
20 remotely.

21 Referring to step 500, the member first
22 establishes a connection to the web server 410 of the
23 broker-to-broker processing system 200 from an end
24 terminal 400/402. The member then enters
25 authentication details including a user name,
26 personal identification number PIN, and secure
27 identity number.

28 At step 502, the authentication details are
29 passed from the web server 410 to the security server
30 470 via the application server 420. The security
31 server 470 can then process the authentication

1 details (see step 504). If the member is not
2 authorized to use the system or has provided
3 incorrect authentication details, a "log-in error"
4 message is generated and sent to the member's end
5 terminal (step 506).

6 If the member is an authorized user and has
7 input his/her authentication details correctly, the
8 security server accepts the login and the application
9 server 420 references the user control information in
10 the front end database 425 to establish whether the
11 member is an originating member or a fulfilling
12 member and what information is to be displayed on the
13 screen of the member's end terminal (see step 508).
14 At step 510, the order manager software 424 of the
15 application server 420 provides the information
16 necessary for the page generator 414 of the web
17 server 410 to display screen information specific to
18 the member at the end terminal 402. This information
19 is then sent to the end terminal for display to the
20 member.

21 The information content of the member's screen
22 is refreshed at predetermined time intervals by the
23 application server 420 referencing the front end
24 database 425 and re-transmitting updated information
25 to the end terminal responsive to a request by the
26 end terminal 400/402. The system is configured such
27 that end terminals request a screen content update at
28 20 second intervals. However, it will be apparent
29 that other time intervals can be used (see step 512).
30 It will be apparent that at any given point in time
31 each member sees a screen display having content

1 defined by the relevant records in the front end
2 database 425 as they existed at the time of the last
3 refresh operation. The content itself is defined by
4 the current order status, while the format may be at
5 least partly defined by member preferences.

6 The order entry frame viewable by originating
7 members enables originating members to enter an
8 order(s) for transmission to a fulfilling member for
9 execution. Items on the order entry frame may
10 include: originating member's investor reference
11 code; transaction type - buy/sell indicator; security
12 identifier; market where security is to be executed;
13 quantity of securities to be executed; price
14 conditions which will govern the execution; timing
15 applicable to the order; duration applicable to the
16 order; identity of the fulfilling member; designation
17 of the execution venue; payment/receive currency;
18 free form message entry area (viewable by both
19 originating member or fulfilling member); intended
20 trade date; intended settlement date; free form text
21 entry area (viewable by originating member only).
22 Further frames allow originating members to enter:
23 identity of the securities account into/from which
24 securities will be delivered/removed; and identity of
25 the cash account into/from which monies will be
26 delivered/debited;

27 Connected members see an orders list on the left
28 hand side of an order management screen. Clicking on
29 an order in the list brings up further details of the
30 order and enables predefined actions to be taken. An

1 originating member can use his end terminal 400, for
2 example, to:

- 3 • enter orders for transmission to the selected
- 4 fulfilling members;
- 5 • request to cancel his/her orders;
- 6 • request to modify attributes of his/her
- 7 orders;
- 8 • respond to requests to cancel initiated by
- 9 the fulfilling member;
- 10 • respond to requests to modify initiated by
- 11 the fulfilling member;
- 12 • request to sign off his/her orders i.e.
- 13 request to initiate the settlement process
- 14 for the fills/executions against his/her
- 15 orders;
- 16 • respond to requests to signoff his/her orders
- 17 initiated by the fulfilling member;
- 18 • receive fill/execution advisories/reports;
- 19 • review the status of his/her orders;
- 20 • provide the information/data required by
- 21 broker-to-broker system to carry out
- 22 settlement management on their behalf;
- 23 • upload material, e.g. notes, pricing
- 24 models/tolls/guidelines, audio clips, video
- 25 clips, to broker-to-broker system central
- 26 electronic information library;
- 27 • search the electronic information library and
- 28 download material;
- 29 • interrogate their billing account with
- 30 broker-to-broker system to obtain information

- 1 such as: transaction fees outstanding to
2 broker-to-broker system, customers'
3 membership fees outstanding, rebates
4 applicable, fee structure agreed with broker-
5 to-broker system, average trader per day
6 (TPD) for week, month, quarter;
- 7 • send messages directly to other broker-to-
8 broker system customer(s) using the secure,
9 guaranteed network facilities;
 - 10 • send messages directly to the customer
11 support system;
 - 12 • search the data repository for securities
13 information;
 - 14 • search the data repository for member
15 information;
 - 16 • obtain information about their use of broker-
17 to-broker system's facilities such as:
18 transaction history (by: time period,
19 security, market, exchange, sector,
20 geographical region), execution quality
21 statistics, volume weighted average price
22 (VWAP), i.e. the price taking into account
23 the size of the transactions executed), delta
24 from best bid or best offer;
 - 25 • security control (authorized names,
26 passwords).

27 Figure 6 is a flow chart illustrating how an
28 order may be placed by a member functioning as an
29 originating member.

1 At step 600 the originating member OM enters, on
2 his/her terminal 400, a security identifier (SID) to
3 identify the security he/she wishes to buy or sell.
4 The OM clicks the "order" button displayed at his/her
5 terminal. The market field is a view-only field,
6 i.e. the user cannot type into this field. An
7 originating member's graphic user interface will only
8 allow securities to be entered in those markets where
9 the broker-to-broker system has signed up fulfilling
10 members as customers since these are clearly the only
11 markets where the order can be transmitted to. A
12 security identifier, SID, along with the market
13 reference code, uniquely identifies the security
14 which the originating member wants to transact. The
15 choice of numbering agency for the SID can be the
16 choice of the originating member. It cannot be
17 sufficiently stressed how easy (ergonomic/user-
18 friendly) determining/looking-up SIDs is using the
19 front end software. At a minimum, the broker-to-
20 broker system will support the following SIDs
21 /numbering agencies: CUSIP (Corp. of United States
22 Instrument Code); ISIN (International Standard for
23 Instrument Numbering); LOCAL (Local Exchanges
24 Identifier); SEDOL (Stock Exchange Daily Order List -
25 - London Stock Exchange Instrument Code); or full
26 company name of the security.

27 At step 604, a message indicating which security
28 is to be bought or sold is sent from the originating
29 member to the order management system server 430 via
30 the web server 410 and the application server 420.
31 The order management server 430 then accesses the

1 settlement management system database 445 via the OMS
2 database 435 and the database link 452 (see step
3 606). The results of the access are passed from the
4 order management server 430 to the application server
5 420.

6 In many cases, securities are listed on multiple
7 exchanges in multiple jurisdictions. The security
8 listed is not a different security, i.e. possession
9 of that security wherever purchased confers the same
10 rights on the holder. Securities such as American
11 Depository Receipts (ADRs) , and Global Depository
12 Receipts (GDRs) are different securities from the
13 underlying and while they may be converted back to
14 the original are still for the purposes of reporting
15 etc, different, distinct securities in their own
16 right. An example of multiple listing is Reuters
17 stock which is listed on other exchanges along with
18 its primary listing on the London Stock Exchange. In
19 these cases the security normally has the same ISIN.

20 If the application server 420 cannot uniquely
21 identify the security to be traded based on the
22 results of the database query, a message is sent from
23 the application server 420 to the end terminal 400
24 containing information on the choices of markets for
25 trading the securities. The end terminal 400
26 displays a choice of markets for the security
27 intended to be traded by the originating member on
28 the screen of the end terminal 400 (see step 608).
29 At step 610, the originating member selects from the
30 choice of markets for trading the security by
31 clicking the screen.

1 Thus, if a SID is entered which causes a
2 multiple match of securities held within the SMS
3 database 445, the full company name of the matching
4 securities along with their traded market of
5 execution will be displayed to the originating
6 member. The originating member then chooses the
7 market he/she wishes to enter the security order on.
8 This will set the market. The requirement for the
9 market to be set is that the broker-to-broker system
10 needs to know to which market the order is being
11 transmitted so that the correct validation checks may
12 be applied, and so that the appropriate conventions
13 and language translations may take place. The order
14 process can then continue as it would have done had
15 the application server been able to uniquely identify
16 the security at step 606 (see below).

17 If the application server 420 can uniquely
18 identify the security selected by the originating
19 member, the originating member is prompted on screen
20 to input particulars of his/her order (see step 612).
21 At step 614, the originating member defines the order
22 by entering specific details of the desired order.
23 Originating members enter information concerning the
24 proposed transaction. In addition, to a valid
25 security identifier, the information includes: the
26 number of securities; whether the order is for a
27 purchase or sale - a buy/sell indicator; valid price
28 conditions for the specific market and security;
29 valid duration and timing of the order; a suggested
30 trade date for the transaction; a suggested
31 settlement date for the transaction; a valid

1 designated fulfilling member; a desired execution
2 venue; the originating member account details for
3 settlement; and valid authenticated identity of the
4 issuing person at the originating member.

5 The originating member may need to keep track of
6 orders entered so that he/she can relate them back to
7 an order(s) which he/she may have been given by a
8 customer. To this end an investor reference code is
9 entered against an order. (Even when the order is
10 for their own-account, i.e. proprietary trading, they
11 may need to track the order). For small originating
12 members, this investor reference code may very well
13 be the same as a customer account. However larger
14 operations may have computer systems.

15 The originating member may enter an alphanumeric
16 string which is their reference code for the order.
17 By convention the client reference code should be
18 used as the reference code. The broker-to-broker
19 system need not in any way interpret/alter this
20 client reference code. The presence or otherwise of
21 this reference code shall have no bearing on how the
22 broker-to-broker system processes the order or its
23 resulting fills. Note also that while this reference
24 code will be electronically attached to the order and
25 all resulting transactions relating to the order,
26 only the originating member will ever view the
27 reference code. This code is not viewable by the
28 fulfilling member. The broker-to-broker system
29 Customer Support personnel, will view the reference
30 code as a field of asterisks, i.e. *****.
31 Even within the broker-to-broker systems it is

1 recommended that this field be rendered non-easily
2 readable to prevent accidental disclosure to the
3 broker-to-broker system technical personnel. This
4 approach helps to ensure that the confidentiality of
5 the investor's details is maintained.

6 The parties involved in the transaction may need
7 to unequivocally identify an order. To this end, the
8 system will generate a unique order reference number
9 which is viewable by all parties, including the
10 customer support desk. This reference code field
11 enables the originating member to put code(s) into
12 his/her own systems so that he/she can reconcile the
13 order and its resulting executions with his/her own
14 systems. This further reference code can be quoted
15 to all parties to the order to unequivocally identify
16 the order in question. For example when the
17 originating member and the fulfilling member are
18 discussing the order, or either party request help
19 from the Customer Support desk, this reference code
20 may allow all parties to rapidly identify the order
21 in question leading to quicker resolution of the
22 problem. This reference code may be visible in human
23 readable form to all parties on the broker-to-broker
24 system.

25 The quantity item allows the user to specify the
26 number of securities to be bought or sold. Short
27 forms to speed up entry and minimize errors are
28 provided on the OMIF, e.g. T for thousands, M for
29 millions, H for hundreds. In addition where lot
30 sizing and/or minimum order sizing applies, broker-
31 to-broker system front end will caution the user

1 whenever the lot size entered is unlikely to be
2 acceptable by the intended market. (The front end
3 software can display the lot sizes applicable).

4 The Buy/Sell indicator specifies the intended
5 direction of the transaction, i.e. whether the order
6 is for a security is to be bought or sold. The
7 indicator represents the direction from the
8 perspective of the originating member, i.e. if the
9 originating member has an order to obtain securities
10 for his/her customer, then the indicator will show
11 BUY. Conversely if the originating member has an
12 order to dispose of securities on behalf of his/her
13 customer, then the buy/sell indicator will show SELL.
14 The price conditions item, allows originating members
15 to specify the price conditions which he/she is
16 prepared to accept when the fulfilling member
17 executes the transaction. The classic price
18 conditions are: sell limit price - i.e. the lowest
19 price that the originating member is prepared to
20 accept in exchange for disposing of his/her
21 securities; buy limit price - i.e. the highest price
22 that the originating member is prepared to pay in
23 exchange for obtaining the securities; sell at market
24 - i.e. the originating member is prepared to receive
25 whatever is the highest current prevailing price in
26 the market; buy at market - i.e. the originating
27 member is prepared to pay whatever is the lowest
28 current prevailing price in the market.

29 Other items comprised in the order entry frame
30 as viewed from an originating member's end terminal
31 400 include: timing applicable to the order; duration

1 applicable to the order; identity of the designated
2 fulfilling member; the commission rate which the
3 fulfilling member will charge to the OM for executing
4 the order; designation of the execution venue;
5 preformatted instructions to the fulfilling member
6 indicating how the order should be traded; free
7 format message area to the FM; free format message
8 area viewable by the OM only; trade date; settlement
9 date; capacity the OM is acting in
10 (agency/principal); payment/receive currency.

11 The system enables the originating member to
12 select among the universe of fulfilling members
13 available for a given security or market. The system
14 need not select a fulfilling member on behalf of the
15 originating member, or change the identity of the
16 originating member's selected fulfilling member on an
17 order. Where an originating member selects a
18 fulfilling member that does not have the capability
19 to trade a particular security or on the particular
20 market in the originating member's order, the
21 originating member chooses a fulfilling member
22 appropriate for that security or market before the
23 order can be further processed.

24 After entering the order details the originating
25 member is prompted to confirm they are correct (see
26 step 614 of Figure 6). If all required information is
27 properly provided, the order is then sent via the
28 broker-to-broker system for verification and for
29 delivery to the fulfilling member.

30 At step 616, a message containing the details of
31 the originating member's order is sent to the order

1 management system server 430. The order processing
2 software 436 of the order management system server
3 430 then stores the details of the order in the OMS
4 database 435 (see step 618). The order management
5 system server 430 can transfer the order information
6 stored in the OMS database 435 to the front end
7 database 425 via the order publishing software 434
8 and the order storage software 423 of the application
9 server 420. At step 620, details of the originating
10 member's order are stored in the front end database
11 525 connected to the application server 420.

12 Operable through the interface's 202,206, for
13 example, using dropdown menus, the conversion service
14 options available to originating members 204 will
15 include: retrieval from data storage and inclusion in
16 the message of appropriate additional securities
17 identification numbers (e.g., the broker-to-broker
18 processing system 200 will be able to provide the
19 applicable ISIN number when given a particular CUSIP
20 number for securities identified for trading in
21 multiple jurisdictions); the reformatting of orders
22 to comply with the customs, dictates or market
23 vernacular of a particular exchange to be used by the
24 designated fulfilling member (e.g. "market held" from
25 a U.S. broker would be converted to "best," the
26 equivalent expression used by broker-dealers in Great
27 Britain); language translation (e.g., English to
28 French) of essential transaction terms (e.g., "buy"
29 or "sell") and the generation and inclusion by the
30 processing system of additional information requested
31 by an originating member 204 that is not an element

1 of a proposed transaction, but offers convenience to
2 the parties, such as display of indicative-only
3 currency conversion data. The processing system 200
4 also offers members an option to include messages
5 which are not be subject to any modification ("free
6 form" messages).

7 Thus, each originating member, and each
8 authorized person acting on behalf of a member, can
9 choose and set the numbering agency which they wish
10 to use to identify securities on their orders. The
11 broker-to-broker system will translate the entered
12 SID into the SID required by the fulfilling broker
13 (and vice versa) and other parties to the
14 transaction. Internally within the broker-to-broker
15 system, securities will be identified by the broker-
16 to-broker system financial instrument identifier
17 (FID). The broker-to-broker system FID is a unique
18 code generated by the broker-to-broker system's data
19 repositories which takes into account for example
20 multiple listings of the same security, (one of the
21 reasons the broker-to-broker system cannot use ISIN
22 internally). The SID entry field allows the
23 originating member to look-up the SID code using the
24 security's text name, or part thereof, e.g. if the
25 user enters "British" in the SID field, an
26 instruction is sent to the order management server to
27 search the SMS database security database and display
28 a list of securities whose full name starts with
29 British. The user then selects the required security.
30 As a convenience, the front end software will
31 remember at least the last 100 SIDs submitted by an

1 originating member, and present that list, together
2 with the text name of the SIDs, to the user on a
3 drop-down list for selection.

4 Using an end terminal to access front end
5 software the fulfilling member can:

- 6 • receive and accept orders from originating members
- 7 for fulfillment;
- 8 • request to cancel his/her orders;
- 9 • request to modify attributes of his/her orders;
- 10 • respond to requests to cancel initiated by the
- 11 originating member;
- 12 • respond to requests to modify initiated by the
- 13 originating member;
- 14 • request to sign off his/her orders (i.e. request
- 15 to initiate the settlement process);
- 16 • respond to requests to sign off his/her orders
- 17 initiated by the originating member;
- 18 • transmit fill/execution advisories/reports to the
- 19 originating member;
- 20 • request to cancel a fill/execution advisory;
- 21 • withdraw any outstanding request which they have
- 22 initiated;
- 23 • review the status of his/her orders to be worked;
- 24 • provide the information/data required by the
- 25 broker-to-broker system to carry out settlement
- 26 management on their behalf;
- 27 • upload material, e.g. notes, pricing models,
- 28 tools, guidelines, audio clips, video clips to the
- 29 broker-to-broker system central electronic
- 30 information library;

- 1 • search the electronic information library and
- 2 download material;
- 3 • interrogate their billing account with the broker-
- 4 to-broker system to obtain information such as:
- 5 transaction fees outstanding to the broker-to-
- 6 broker system, customer fees outstanding, rebates
- 7 applicable, fee structure agreed with the broker-
- 8 to-broker network administrator, average traders
- 9 per day TPD for week, month, quarter;
- 10 • send messages directly to other members using the
- 11 secure, guaranteed the broker-to-broker system
- 12 network facilities;
- 13 • send messages directly to the customer support
- 14 system;
- 15 • search the data repository for securities
- 16 information;
- 17 • search the data repository for member information;
- 18 • obtain information about their use of the broker-
- 19 to-broker system's facilities such as: transaction
- 20 history (by: time period, security, market,
- 21 exchange, sector, geographical region);
- 22 • effect security control (authorized names,
- 23 passwords).

24 Figure 7 illustrates how a fulfilling member
25 becomes aware of orders placed with him and may
26 accept or reject them. A fulfilling member may
27 become aware of new orders placed with him when he
28 establishes a connection to the broker-to-broker
29 system 200 or when the screen content of the end
30 terminal 402 is refreshed in a period after new order

1 particulars have been transferred to the front end
2 database 425.

3 Referring to step 700, the fulfilling member
4 sees all new orders along with all pending orders on
5 an orders list on the left-hand side of his screen.
6 The fulfilling member can select one and view the
7 order particulars in detail.

8 If the fulfilling member elects to view the
9 particulars of a new order placed with him, the
10 fulfilling member will click on the order within the
11 order list whereby the application server 420
12 references the front end database 425 and the order
13 manager software 424 supplies details of the order to
14 the end terminal 402 via the page generator 414 in
15 the web server 410. Fields presented on this screen
16 are read-only except where otherwise noted. The order
17 acceptance frame enables the fulfilling member to
18 accept an order(s) for fulfillment. Items on the
19 order acceptance frame may include: means for the
20 fulfilling member to accept or reject the order;
21 originating member's order reference code;
22 transaction type - buy/sell indicator; security name;
23 security identifier; market where security is to be
24 executed; quantity of securities to be executed;
25 price conditions which will govern the execution;
26 designation of the execution venue; payment/receive
27 currency; trade date; settlement date; and commission
28 rate; free form message area. Other frames allow the
29 fulfilling member to provide the identity of the
30 securities account into/from which securities will be

1 delivered/removed; and identity of the cash account
2 into/from which the monies will be delivered/debited;

3 When an order is transmitted to the fulfilling
4 member, an alert mechanism, visual and audible,
5 prompts the fulfilling member that there is an order
6 to be dealt with. Failure to respond to the alert
7 within a pre-determined time-period may cause the
8 order to be automatically cancelled and the
9 originating member so notified. This time-period for
10 acknowledgement is configurable on the following
11 basis: system wide; per market; per fulfilling
12 member.

13 Electronic acknowledgement of the order allows
14 the broker-to-broker system 200 to send a positive
15 indication back to the originating member that the
16 order has been delivered to the designated fulfilling
17 member and is being considered for acceptance. This
18 order acknowledgement does NOT mean that the
19 fulfilling member has agreed to fulfil the order,
20 i.e. no contract has yet been formed between the
21 originating member and the fulfilling member. Once an
22 order has been acknowledged, details of the order are
23 placed onto the FMIF pending orders manager screen.
24 The order having been acknowledged, all parties now
25 "know" that the order has been seen by the fulfilling
26 member and could be fulfilled. The fulfilling member
27 now has to accept, reject, or request modification to
28 the order within a configurable time-period. This
29 acceptance time-period is configurable on the
30 following basis: system wide; per market; per
31 fulfilling broker.

1 The Buy/Sell indicator specifies the intended
2 direction of the transaction to the fulfilling
3 member. The Security Identifier, SID, along with the
4 market reference code, uniquely identifies the
5 security which the fulfilling member is to transact
6 on behalf of the originating member. Automatic
7 reformatting and conversion/translation functions of
8 the broker-to-broker function may mean that the
9 content of certain fields viewed by the fulfilling
10 member is different to that input by the originating
11 member. The format and content type of the subject
12 matter viewed at an end terminal is
13 selected/configured by the member who uses the end
14 terminal. For example in the case of a fulfilling
15 member, the choice of numbering agency for the SID is
16 primarily the choice of the fulfilling member. The
17 broker-to-broker processing system will map the
18 originating member's SID into the form required by
19 the fulfilling member. At a minimum, the broker-to-
20 broker system will support the following SIDs
21 /numbering agencies: CUSIP, ISIN, SEDOL, LOCAL
22 (Local Exchange Identifier) and full company name.
23 Each fulfilling member, and each authorized person at
24 the fulfilling member's premises, can choose and set
25 the numbering agency which they wish to use to
26 identify securities on the orders which they receive.
27 The broker-to-broker system will, if required,
28 convert the originating member's SID into the SID
29 required by the fulfilling member and other parties
30 to the transaction. Internally, i.e. within the
31 broker-to-broker system, securities will be

1 identified by the broker-to-broker system FID. The
2 broker-to-broker system FID is a unique code
3 generated by the broker-to-broker system 's data
4 repository which takes into account for example
5 multiple listings of the same security.

6 A security to be transacted by a fulfilling
7 member is uniquely identified by means of the
8 security identifier SID and the market. The quantity
9 item specifies the number of securities to be bought
10 or sold by the fulfilling member. Where lot sizing
11 and/or minimum order sizing applies, the FMIF front
12 end software expresses the quantity as an absolute
13 quantity and as a number of lots to sell/buy. The
14 price conditions item indicates the price conditions
15 the originating member is prepared to accept when the
16 fulfilling member executes the transaction. Examples
17 of price conditions are: sell limit price - i.e. the
18 lowest price that the originating member is prepared
19 to accept in exchange for disposing of his/her
20 securities; buy limit price - i.e. the highest price
21 that the originating member is prepared to pay in
22 exchange for obtaining the securities; sell at market
23 - i.e. the originating member is prepared to receive
24 whatever is the highest current prevailing price in
25 the market; buy at market - i.e. the originating
26 member is prepared to pay whatever is the lowest
27 current prevailing price in the market.

28 Other items accessible to the fulfilling broker
29 include: identity of the fulfilling member; the
30 commission rate which the firm will charge to the
31 originating member for executing the order;

1 designation of the execution venue; preformatted
2 instructions from the originating member indicating
3 how the order should be traded; free format message
4 area from the originating member; identity of the
5 originating member who routed the order; trade date;
6 settlement date; capacity the originating member is
7 acting in; payment/receive currency; transaction
8 management; intra member messaging; interface to the
9 broker-to-broker system electronic library.

10 The fulfilling member reviews the information
11 for completeness and accuracy. If the need arises he
12 can contact the originating member via a
13 communication means of the broker-to-broker system.
14 The fulfilling member can then determine in its sole
15 discretion whether to accept or reject the order. The
16 fulfilling member is presented with on-screen buttons
17 inviting him/her to accept or reject the order (see
18 step 702).

19 When a fulfilling member rejects an order, the
20 originating member is sent an order reject message
21 together with an indication of the reason for the
22 rejection. That is, the fulfilling member is prompted
23 to indicate a reason for rejecting the order and an
24 order reject message including the reason is sent
25 from the end terminal 402 to the order management
26 system server 430 via the web server 410 and the
27 application server 420 (see step 704). At step 706,
28 the order record in the OMS database 435 is accessed
29 to add an indication that the order has been
30 rejected. An indication that the order has been
31 rejected is then replicated to the front end database

1 425 via the order publisher software 434 and the
2 order storage software 423 of the order management
3 system server 430 and the application server 420,
4 respectively (see step 708). The status of the order
5 within the broker-to-broker system is thus changed to
6 "rejected". Step 710 illustrates that the originating
7 member's screen content is changed to indicate "order
8 rejected" instead of "order placed" next time the
9 screen content is updated.

10 When a fulfilling member accepts an order, a
11 contract is now deemed to exist between the
12 originating member and the fulfilling member. The
13 contract is governed by the terms of the order, as
14 specified in the fields of the order, the terms of
15 the fulfilling members agreements with the broker-to-
16 broker system, and is subject to the regulations of
17 the fulfilling member's jurisdiction, e.g. the member
18 agrees to the desired quantity, price conditions,
19 delivery conditions and execution venue specified by
20 the originating member. Thus, if the fulfilling
21 member accepts an order, an order acceptance message
22 is sent from the end terminal 402 to the order
23 management system server 430 (see step 712). The
24 order processor 436 of the order management system
25 server 430 then updates the relevant order record in
26 the OMS database 435 to indicate acceptance of the
27 order by the fulfilling member (see step 714). An
28 indication that the order has been accepted is then
29 transferred from the OMS database 435 to the front
30 end database 425 by means of the order publishing
31 software 434 and the order storage software 423 (see

1 step 716). Next time the screen content of the
2 originating member (or the fulfilling member)
3 terminal is updated by one of the periodic references
4 to the front end database 425, the display of the end
5 terminal 400 (or 402) will indicate an "order pending
6 fill" status to indicate acceptance (718).

7 Once the originating member is notified of
8 acceptance it can be assumed that the fulfilling
9 member will seek to execute the order according to
10 the terms specified. No item on the order may now be
11 altered without requesting the fulfilling member to
12 positively accept the changes requested.

13 In the case that the order is accepted, the
14 fulfilling member seeks to execute the order on the
15 specified market satisfying the terms of the order.
16 The fulfilling member and the originating member can
17 communicate through the broker-to-broker system or
18 through any other means they may choose as to the
19 status of fulfillment of the order and any
20 modifications thereto. When the members agree that
21 the order has been fulfilled or the unfulfilled
22 portion cancelled, information concerning the
23 transaction is sent to the originating member for
24 review. The agreed transaction can then be submitted
25 by both members to the settlement management portion
26 of the system which then generates settlement
27 instructions, based entirely on transaction details,
28 settlement account and preference information
29 previously provided by members, for transmission to
30 the settlement agents designated exclusively by the
31 respective members.

1 Figure 8 is a flow chart illustrating how the
2 originating member is made aware that the fulfilling
3 member has executed an order placed with him.
4 Referring to the step designated 800, the fulfilling
5 member selects the order he has executed from the
6 list displayed on the screen of end terminal 402. To
7 provide the fulfilling member with more details of
8 the order he has selected, the application server 420
9 references the front end database 425 and content
10 relevant to the order selected is supplied to the end
11 terminal 402 via the web server 410 (see step 802).
12 The fulfilling member can then enter execution data
13 (see step 804). At step 806, the execution data
14 supplied by the fulfilling member are sent to the
15 order management system server 430 and are recorded
16 by the order processor 436 in the OMS database 435
17 (see step 808). The execution data are subsequently
18 transferred to the relevant record in the front end
19 database 425 via the order publishing software 434
20 and the order storage software 423 (see step 810).
21 The "executed" status of the order is then indicated
22 on the displays of both the originating member and
23 the fulfilling member end terminals next time the
24 screen content is refreshed.

25 If the order has been fully executed by the
26 fulfilling member, either the originating member or
27 the fulfilling member may initiate a settlement
28 procedure (see step 810). If on the other hand the
29 order has been partially executed by the fulfilling
30 member, the fulfilling member can execute the
31 remainder later. If the fulfilling member executes

1 the remainder of the order later (see step 812), the
2 process from step 800 is followed to report
3 completion of order to the originating member. In
4 cases where the fulfilling member does not execute
5 the remainder of the order that day, the order status
6 is preserved until the next suitable day.

7 Figure 9 is a flow chart illustrating how
8 settlement instructions are generated and issued by
9 the broker-to-broker computer system. Either party
10 can initiate the settlement procedure, however in
11 this example the procedure is initiated by the
12 fulfilling member. Referring to step 900, the
13 fulfilling member selects the executed order for
14 which he wishes to initiate settlement proceedings by
15 clicking on it. The fulfilling member's screen
16 prompts him/her to "request sign-off" (see step 902).
17 When a fulfilling member selects the request sign-off
18 button, he/she is prompted to enter settlement
19 information, including data identifying the
20 settlement account of the fulfilling member (see step
21 904). At step 906, the settlement information
22 provided by the fulfilling member is sent from the
23 end terminal 402 to the order management system
24 server 430 from where it is provided to the
25 settlement management system server 440. The order
26 management system server 430 transfers the fulfilling
27 member's settlement information to the SMS database
28 445 via the OMS database 435 and the database link
29 452. The OMS server 430 makes a database stored
30 procedure call into the SMS database 445 passing in
31 the details of the relevant trade, responsive to

1 which the OMS server 430 is provided with the
2 calculated commissions, and any local market charges,
3 taxes or levies (see step 908). At step 910 the
4 calculated settlement criteria are sent directly to
5 the fulfilling member terminal 402 where the
6 fulfilling member can review them and either accept
7 or reject them.

8 If the fulfilling member accepts the settlement
9 criteria, a message containing the accepted
10 settlement particulars is sent to the order
11 management system server 430 which generates a
12 message changing the status at the order management
13 system server 430. Certain particulars, e.g. average
14 prices and security settlement account information is
15 stored in the OMS database 435. These settlement
16 particulars are transferred into the front end
17 database 425 via the order publisher software 434 of
18 the OMS server 430 and the order storage software 423
19 of the application server 420. The status of the
20 order is thus updated to be "sign off requested"
21 which will be displayed to the originating member and
22 fulfilling member the next time screen content on the
23 terminals 400, 402 is refreshed (912).

24 Referring to step 914, the originating member
25 clicks on the "sign-off requested" button and is
26 prompted to accept/reject the request. The OM can
27 accept and provide settlement information, including
28 information identifying the originating member's
29 settlement account (see step 915). After the
30 originating member has entered his/her settlement
31 information, the settlement information is sent to

1 the order management system server 430 for processing
2 based, for example, on for example average price
3 information provided by the fulfilling member (see
4 step 916). At step 918, the order management system
5 server accesses the SMS database 445 by means of a
6 stored procedure call passing the trade details into
7 the SMS database 445. In response, calculated
8 settlement criteria relating to the originating
9 member are supplied from the SMS database to the OMS
10 database 435 via the database link 452 (see step
11 918). The order management system server 430
12 transfers the calculated settlement criteria of the
13 originating member to the end terminal 400 where the
14 originating member can review and either accept or
15 reject them (see step 920).

16 If the settlement criteria are accepted by the
17 originating member, a message containing the accepted
18 particulars is sent from the end terminal 400 back to
19 the order management system server 430 and the
20 accepted particulars are stored in the OMS database
21 435 (see step 922). The front end database 425 is
22 updated by means of the order publisher software 434
23 and the order storage software 423 in the usual way,
24 after which update the screen content can reflect the
25 new status of "signed off".

26 Once both the originating member and the
27 fulfilling member have accepted the settlement
28 criteria of a given trade, the order management
29 server creates two trade records (see step 924). One
30 trade record corresponds to a buy transaction and is
31 associated with the appropriate one of the

1 originating or fulfilling members. The other trade
2 record corresponds to a sell transaction and is
3 associated with the other one of the originating or
4 fulfilling members. The two trade records are stored
5 in the SMS database 445 via the OMS database 435 and
6 the database link 452.

7 The settlement management server 440 polls the
8 SMS database 445 at predetermined intervals to find
9 trade records which require processing. Having
10 detected the trade records the SMS server 440
11 validates certain order attributes (e.g. security,
12 market, market listing, trade date, price, settlement
13 date, broker identities, commission rates and default
14 charges) against static data held in the SMS database
15 445. The initiation of the settlement procedures is
16 undertaken by the generation of agent instructions
17 during the trade verification process. Referring to
18 step 926, the trade processor 442 of the settlement
19 management server 440 processes the settlement
20 management information of that transaction to
21 generate settlement instructions (see step 926).

22 In this way, generic trade instructions are
23 generated and associated with a given medium that
24 identifies how the message will be sent to the
25 clearing agent. The settlement management sub-system
26 thus derives the company's stock and cash settlement
27 agent and the client's stock and cash settlement in
28 order to ascertain whether the trade will be settled
29 either against or free of payment. Considerable
30 flexibility is available for defining settlement
31 preferences for the company and its counterparties.

1 Settlement instructions can be established and varied
2 for a wide range of criteria such as particular
3 markets, instrument types, individual stocks, and
4 currencies. A first settlement instruction message
5 is sent to the settlement agent designated by the
6 originating member and a second settlement
7 instruction message is sent to the settlement agent
8 designated by the fulfilling member. The settlement
9 instructions may be transmitted to settlement agents
10 in, for example, SWIFT format messages 443 or by fax
11 message 444. (SWIFT is an international standard
12 recognized by banks for sending messages to make
13 payments of cash and securities.) It will be
14 apparent that criteria and criteria acceptance
15 messages may also be routed between the originating
16 member, the fulfilling member via indirect routes.
17 For example, the fulfilling member's indication of
18 acceptance may instead be sent directly to the
19 originating member along with criteria for acceptance
20 by him/her. Both indications of acceptance can then
21 be returned to the processing system together. In
22 the context, of receiving and sending information
23 items the terms "to originating party/fulfilling
24 member" and "from originating party/fulfilling
25 member" should be construed accordingly.

26 Thus, appropriate settlement instructions are
27 generated automatically by the settlement management
28 part 440,445 of the broker-to-broker processing
29 system 200 for all parties to the trade and sent to
30 the relevant settlement agents. Settlement
31 instructions relating to agreed settlement

1 particulars are issued from a single source and at an
2 appropriate point in time. In this preferred
3 embodiment, settlement instructions are issued to the
4 settlement agent of the originating member and the
5 settlement agent of the fulfilling member at the same
6 time.

7 In preferred embodiments the originating member
8 interface OMIF 202 includes an application program
9 interface, referred to herein as BIAPI, designed to
10 allow the broker-to-broker members to interface their
11 systems directly into broker-to-broker system 200
12 eliminating and minimizing the need to key data into
13 order entry screens, and keying data about fills,
14 positions etc into their systems.

15 There are two forms of BIAPI: BIAPI-F - File
16 based interface; and BIAPI-I - Interactive interface.

17 The file based API, BIAPI-F, is designed to
18 enable the widest, simplest and quickest system
19 interconnect possibilities, whilst the interactive
20 API, BIAPI-I, is designed to enable hi-performance,
21 functionally rich system interconnects to be created.

22 Technically, BIAPI-F consists of text files
23 containing comma separated, quoted tag and value
24 records terminated by an end-of-line character. These
25 files may be read or put from/into the file-space of
26 the terminal 400 client, e.g.

27 "SNA =CUSIP", (Security Numbering Agency)
28 "SID =0123456A", (Security Identifier)
29 "QTY =500", (Quantity)
30 "PXTYPE =LIMIT", (Pricing Type)
31 "LIMIT =10.5", (Limit Order Level)

1 "PAYCUR =EUR" (Payment Currency)

2 BIAPI-I may be implemented as a component, which
3 exposes methods that are then invoked by the host
4 program.

5 The functionality provided through either BIAPI-
6 F, or BIAPI-I, will include: send orders for
7 execution; receive fill/execution notification;
8 receive order status; receive settled position(s) and
9 balance(s) information; receive open positions(s) and
10 balance(s) information; receive their account
11 information, i.e. electronic billing from the broker-
12 to-broker system; send settlement account(s)
13 information to the broker-to-broker system; receive
14 transaction history information.

15 Through BIAPI-I, the originating members are
16 able to carry out all of the functions available
17 through BIAPI-F, and the following additional set of
18 functionality: session control; log on/off; manage
19 transactions; request cancellation; request
20 modification; request sign-off; accept/reject
21 cancellation requests; accept/reject modification
22 requests; accept/reject sign-off requests; withdraw
23 an initiated request; query the interface for order
24 status; security lookup; request for quote; request
25 reports/data with filter specifications; transaction
26 history; open position(s) and balance(s) information;
27 settled positions(s) and balance(s) information;
28 their account information, i.e. electronic billing
29 from the broker-to-broker system.

30 The API of the FMIF 206, is designed to allow
31 the broker-to-broker system's customers to interface

1 their systems directly into the broker-to-broker
2 system eliminating and minimizing the need to key in
3 order data and keying data about fills, positions
4 etc. into the fulfilling member's trading systems.

5 There are two forms of API for the FMIF 206
6 SELAPI: SELAPI-F - File based interface; and SELAPI-I
7 - Interactive interface.

8 The file based API, SELAPI-F, is designed to
9 enable the widest, simplest and quickest system
10 interconnect possibilities, whilst the interactive
11 API, SELAPI-I, is designed to enable hi-performance,
12 functionally rich system interconnects to be created.

13 SELAPI-F comprises text files containing comma
14 separated, quoted tag and value records terminated by
15 an end-of-line character. These files may be read or
16 put from/into the file-space of the client terminal
17 402, e.g. "SNA =CUSIP""SID =0123456A","QTY
18 =500","PXTYPE =LIMIT","LIMIT =10.5","PAYCUR =EUR"
19 etc.

20 SELAPI-I comprises a component, which exposes
21 methods that are then invoked by the host program.

22 The functionality provided through either
23 SELAPI-F, or SELAPI-I, will include: receive orders
24 for execution; transmit fill/execution notification;
25 receive settled position(s) and balance(s)
26 information; receive open positions(s) and balance(s)
27 information; receive their account information, i.e.
28 electronic billing from the broker-to-broker system;
29 send settlement account(s) information to the broker-
30 to-broker system; receive transaction history
31 information.

1 Through SELAPI-I, the fulfilling members are
2 able to carry out all of the functions available
3 through SELAPI-F, and the following additional set of
4 functionality: session control; log on/off; manage
5 transactions; request cancellation; request
6 modification; request sign off; accept/reject cancel
7 requests; accept/reject modification requests;
8 accept/rejection sign off requests; withdraw an
9 initiated request; request cancellation of fills;
10 query the interface for order status; security
11 lookup; respond to quote requests; request
12 reports/data with filter specifications; transaction
13 history; open position(s) and balance(s) information;
14 settled positions(s) and balance(s) information; and
15 their account information, i.e. electronic billing
16 from the broker-to-broker system.

17

18 TECHNOLOGY STANDARDS

19 The broker-to-broker network and interfaces
20 adhere to as many standards as possible for the
21 following reasons: lower overall cost due to
22 economics of mass-market production and sales;
23 standards based technologies tend to have fewer
24 problems/bugs, i.e. more dependable; easier to
25 resource personnel; leverage of existing technology
26 intellectual property; ability to stay current with
27 and leverage technological changes; and ease of
28 interfacing to other systems both internally and
29 externally to the broker-to-broker network.

30 In this embodiment, the standards chosen by the
31 broker-to-broker system are as follows:

1 Netscape Navigator, Microsoft Internet Explorer
2 - for Web browsers
3 HTML 4, DHTML, SSL, JavaScript - for Web front-
4 end programming
5 Netscape, Apache - for Web servers
6 BEA Systems WebLogic - for application servers
7 C/C++, Java, Perl - for application processing
8 engines
9 UNIX - SOLARIS, HP/UX, LINUX - for application
10 processing engines
11 Windows NT - for in-house desktop technology
12 ORACLE RDBMS - for databases
13 SUN Microsystems, COMPAQ - for application
14 hardware
15 COMPAQ - for in-house desktop hardware
16 STRATUS - for non-stop 100% dependability
17 hardware
18 TCP/IP - for base level networking protocols
19 SWIFT - for payment instruction protocol
20 FIX - for order management protocol
21 Microsoft Office - for document creation
22 Adobe Acrobat - for document distribution
23 RSA Dynamics SecurID - for authentication
24 controls
25 The broker-to-broker system transaction rules,
26 legalities and processes.
27 Of course, the functionality embodying the
28 present invention can be achieved with any suitable
29 combination of network elements and protocols.

1 In this embodiment the broker-to-broker
2 processing system 200 includes the following
3 characteristics:

4 The processing system 200 does not change the
5 sequencing of orders received from the originating
6 member 204. There is no prioritizing of orders from
7 any given originating member 204, nor any
8 prioritizing of orders among originating member 204.
9 The processing system does not generally change the
10 sequencing of fulfillment advisories transmitted from
11 a fulfilling member 208 back to an originating member
12 204.

13 The processing system 200 enables the
14 originating member 204 to select from a plurality of
15 fulfilling members 208 available for a given security
16 or market. Thus, the processing system 200 does not
17 in this example select a fulfilling member 208 on
18 behalf of the originating member 204, or change the
19 identity of the originating member's 204 selected
20 fulfilling member 208 against an order. Where an
21 originating member 208 selects a fulfilling member
22 204 that does not have the capability to trade a
23 particular security or on the particular market in
24 the originating member's 204 order, the originating
25 member 204 may be required to choose a fulfilling
26 member 208 appropriate for that security or market
27 before the order can be further processed by the
28 processing system 200.

29 Originating member's 204 may be able to direct
30 and monitor the nature of the fulfillment of their

1 orders through various-mechanisms, initially
2 including at least the following:

3 The originating member 204 may be able to
4 designate and set as a default the execution venue
5 which the fulfilling member 208 may be required to
6 use when fulfilling the order, for example, on an
7 exchange, off-exchange or a crossing system, for a
8 particular security or member.

9 In more developed electronic markets, every
10 executed trade reported by the fulfilling member 208
11 will be time stamped and marked with the best
12 official bid and offer positioned in that market, to
13 enable originating members 204 to compare the
14 execution quality received against market pricing.

15 The processing system 200 may be able to
16 generate reports of transactions conducted using the
17 broker-to-broker system to help brokers comply with
18 their regulatory reporting obligations.

19 Settlement of a trade between the originating
20 member and its investor, or between the fulfilling
21 member and the street, can be completely within the
22 province of the member. The system need not generate
23 settlement instructions, confirmations or other forms
24 of advisories or messages to either member's
25 investors or counter-parties.

26 All transactions that are initiated by
27 communications made through the system may be
28 executed by the members to the transaction in a
29 market and in a manner in which they are qualified to
30 conduct such business. The system is a facility

1 through which information relating to securities
2 orders may be transmitted.

3 The conversion service features of the system
4 may provide opportunities for increased efficiency of
5 cross-border and inter-market transactions and thus
6 contribute to broader access to securities markets
7 worldwide, in particular because the system may offer
8 those capabilities at lower variable costs to members
9 who have previously been less able to fully take
10 advantage of advanced technologies to overcome the
11 costs and other burdens arising from the numerous
12 additional variables associated with international
13 securities transactions. Accordingly, the use of the
14 system by members may further increase the
15 transparency of international securities markets and
16 afford to investors through their brokers increased
17 liquidity and efficiency in conducting securities
18 transactions.

19 A flat annual fee may be charged to originating
20 members and, only after a certain number of
21 transactions, an excess usage fee may be charged to
22 originating and fulfilling members whose use of the
23 system results in the generation of a high volume of
24 settlement instructions.

25 The excess usage fee is not dependent upon the
26 completion of any securities transaction. The
27 settlement service is cost-effective and more error-
28 free given the symmetry of data entered onto the
29 system by the brokers conducting a transaction. The
30 excess usage fee is intended to fairly compensate the
31 broker-to-broker network operator based on a

1 proportional accounting for substantial usage of the
2 system indicated by a greater than anticipated volume
3 of transactions submitted for settlement.
4 Accordingly, unless a system failure or error occurs,
5 no rebate may be provided for an excess usage fee if
6 a transaction submitted for settlement does not in
7 fact close. Indeed, the network operator may reserve
8 the right to charge an additional excess usage fee
9 for further settlement processing when a failure to
10 settle is due to member error. The network
11 operator's receipt of excess usage fees when a
12 certain number of settlement instructions are
13 generated would not cause the network operator to be
14 effecting transactions in securities because the
15 network operator has no substantive role in whether
16 any trade is executed and submitted for settlement,
17 and once submitted, whether the trade successfully
18 closes.

19 The payment structure contemplated by the
20 network operator reflects the intent for the system
21 to be rigorously neutral as to the effectuation of
22 transactions and to allocate among users their
23 respective share of the costs of operating the system
24 at the service levels demanded in a commercially
25 rational manner. The fees charged by the operator
26 may be based on the operating overhead and the
27 expenses of maintaining the system, plus recapture of
28 start-up costs and a market driven profit.

29 While this invention has been described in
30 connection with what is presently considered to be
31 the most practical and preferred embodiment, it is to

1 be understood that the invention is not limited to
2 the disclosed embodiment, but on the contrary, is
3 intended to cover various modifications and
4 equivalent arrangements included within the spirit
5 and scope of the appended claims.

6
7 EXAMPLE OF ALTERNATIVE EMBODIMENT

8 Figure 10 illustrates an example of a further
9 computer system embodying the present invention.
10 BINET 1010 is a secure private network using Internet
11 technology even though it itself need not be
12 implemented on the Internet. Provided appropriate
13 security measures are implemented, such as use of the
14 public key infrastructure (PKI), BINET may comprise
15 the internet in some embodiments, as will be
16 explained below. Thus, this broker-to-broker system
17 can leverage the massive strides taking place in web
18 and communications technology at low cost. Of note
19 is that BINET 1010 interfaces into the core of the
20 broker-to-broker system using the FIX protocol,
21 providing the broker-to-broker system with the
22 ability to replace/upgrade BINET 1010 without unduly
23 impacting the main the broker-to-broker system
24 processing core, and also providing a standard
25 interface into which the broker-to-broker system's
26 larger customers can directly interface their
27 systems.

28 Order Management System (OMS) 1000 may be a UNIX
29 application, written in C++, an Oracle RDBMS, and
30 Microsoft Windows NT GUI front-end which acts as the

1 transaction manager keeping track of orders, their
2 status and directing the sequence of processing

3 SELNET 1012 is a secure private network using
4 Internet technology even though it itself need not
5 necessarily be implemented on the Internet. Where
6 appropriate security measures are put in place (e.g.
7 PKI), SELNET may comprise the internet. In this way
8 the broker-to-broker system can leverage the massive
9 strides taking place in web and communications
10 technology at low cost.

11 SMS, Settlement Management System 1006, may be a
12 UNIX application, written in C/C++, an Oracle RDBMS,
13 and Microsoft Windows NT GUI front-end, which
14 controls the settlement of the proceeds of the
15 transactions. Of key importance is SMS's 1006
16 ability to automatically generate SWIFT settlement
17 instructions 1080 for all parties to the trade.
18 (SWIFT is an international standard recognized by
19 banks for sending messages to make payments of cash
20 and securities.)

21 Settlement and Custody 1008 may be external
22 agents (to the broker-to-broker system) who actually
23 carry out the physical act of delivering/receiving
24 cash and securities

25 CSS, Customer Service System 1004, may be a UNIX
26 and NT suite of applications which collects and
27 reports status from other the broker-to-broker system
28 systems, tracks the relationships with the broker-to-
29 broker system's customers and provides access into
30 the broker-to-broker system's systems to enable
31 service personnel to correct problems.

1 CDR 1002, Central Data Repository, may be
2 implemented with an ORACLE RDBMS operating under
3 HP/UX hosted on 100% non-stop STRATUS Continuum
4 computer hardware. The CDR 1002 is in this embodiment
5 the single source from which all of the broker-to-
6 broker system's sources static data, and into which
7 all of the broker-to-broker system's systems send
8 status messages. It will be apparent that the key
9 information flows of the computer system of Fig. 10
10 are similar to those described with reference to
11 Figure 2 and much of the same advantageous functions
12 can be achieved.

13 Figure 11 illustrates how an end terminal in the
14 computer system of Figure 10 is connected to the
15 broker-to-broker system. The end terminal shown in
16 Figure 10 is generally similar to that in Figure 3
17 and like reference numerals designate like features.

18 In this modification, the communication
19 interface 118 is a local area network (LAN) card to
20 provide a data communication connection to a
21 compatible LAN. In alternative embodiments, hard-
22 wired circuitry may be used in place of or in
23 combination with software instructions to implement
24 the invention. Thus, embodiments of the invention
25 are not limited to any specific combination of
26 hardware circuitry and software. In any such
27 implementation, communication interface 118 sends and
28 receives electrical, electromagnetic or optical
29 signals that carry digital data streams representing
30 various types of information.

1 Network link 120 typically provides data
2 communication through one or more networks to other
3 data devices. For example, network link 120 may
4 provide a connection through local network 122 to a
5 host computer 124 or to data equipment operated by an
6 Internet Service Provider (ISP) 126. ISP 126 in turn
7 provides data communication services through the
8 worldwide packet data communication network, now
9 commonly referred to as the "Internet" 128. Local
10 network 122 and Internet 128 both use electrical,
11 electromagnetic or optical signals that carry digital
12 data streams in accordance with well established
13 protocols such as the TCP/IP protocol suite. The
14 signals through the various networks and the signals
15 on network link 120 and through communication
16 interface 118, which carry the digital data to and
17 from computer system 400/402, are exemplary forms of
18 carrier waves transporting the information.

19 Computer system 400/402 can send messages and
20 receive data, including program code, through the
21 communications link 120, and communication interface
22 118. In this Internet example, a server 130 might
23 transmit a requested code for an application program
24 through Internet 128, ISP 126, local network 122 and
25 communication interface 118. In accordance with this
26 embodiment of the invention, such downloaded
27 instruction code provides for implementing a broker-
28 to-broker system as described herein.

29 The received code may be executed by processor
30 104 as it is received, for example by using a browser
31 application installed upon the end terminal, and/or

1 stored in storage device 110, or other non-volatile
2 storage for later execution. In this manner,
3 computer system 400/402 may obtain application code
4 in the form of a carrier wave.

5 It will be apparent that any suitable end
6 terminal device access network arrangement can be
7 used to facilitate communication between a member and
8 the broker-to-broker processing system 200. For
9 example, alternative end terminals include mobile
10 computers and personal digital assistants equipped
11 for operation according to established communication
12 protocols for internet based technologies, such as
13 Wireless Application Environment WAE protocols and
14 later equivalents. Wireless links may be implemented
15 on wireless communication systems based on techniques
16 such as time division multiple access (TDMA),
17 frequency division multiple access(FDMA), space
18 division multiple access(SDMA), code divisional
19 multiple access (CDMA) and hybrids thereof. Some
20 such systems are established and widely used,
21 examples being AMPS and GSM, other such systems are
22 under deployment, such as the Mobile Station-Base
23 Station Compatibility Standard for Dual-Mode Wideband
24 Spread Spectrum Cellular System (IS95) and the
25 Universal Mobile Telecommunications System (UMTS).

26 Figures 12 and 13 illustrate alternative access
27 networks BINET and SELNET which may be used by
28 originating and fulfilling members, respectively, to
29 connect to the embodiment of Figure 10.

30 Referring to FIG. 12, the originating member
31 interface OMIF 202 may replaced with an access

1 network BINET connecting the originating member with
2 the broker-to-broker processing system 200 and
3 providing him/her with the ability to interact with
4 other customers. The access network BINET comprises:
5 an end terminal 1230 through which the originating
6 member interacts with broker-to-broker processing
7 system 200; a computer-to-computer applications
8 programmer interface, BINET-API, which the
9 originating member may use to interface his/her
10 systems directly into the broker-to-broker system
11 network so as to eliminate rekeying of data. BINET-
12 API can be made available, for example, as an
13 interactive interface, BIAPI-I, or as a file/batch
14 interface, BIAPI-F; a secure, encrypted, hi-speed,
15 reliable and guaranteed QOS TCP/IP network
16 connection, BICON, which links BINET sessions at the
17 originating member location into the broker-to-broker
18 system.

19 Referring to Figure 13, the fulfilling member
20 interface may be replaced by an access network SELNET
21 connecting fulfilling members to the broker-to-broker
22 processing system. Similarly, a computer-to-computer
23 application program interface, SELNET-API can be used
24 by the fulfilling member to interface his/her systems
25 directly into the broker-to-broker system 200.
26 SELNET-API is available as an interactive interface,
27 or as a file/batch interface; a secure, encrypted,
28 hi-speed, reliable and guaranteed QOS TCP/IP
29 network connection, which links SELNET sessions at
30 the fulfilling member's location into the broker-to-
31 broker system.

1 BIWEB/SELWEB operate on BINET and SELNET,
2 respectively, and comprise two major parts, the
3 front-end or GUI which the fulfilling members
4 interact with, and the application core which
5 processes the data entered by the members.

6 The front-end may be built using the following
7 standards: Netscape Navigator 4.5., Microsoft
8 Internet Explorer 4.01; HTML 4 + JavaScript; JAVA;
9 128 Bit SSL.

10 The application core may be built using the
11 following standards: Netscape Commerce or Apache
12 Server; CGI with Perl and C/C++; Java servelets; Java
13 Servelet Pages; Application Server; SUN Solaris or
14 Linux Operating System; Oracle Database; Tellurian
15 Secure Sockets.

16 A target technical environment at the fulfilling
17 member may be Netscape Navigator 4.5 or Microsoft
18 Internet Explorer 4.01, higher versions may be used.

19 Cookies, small files stored on the user's local
20 hard-drive, are commonly used by web based
21 applications to store data about the user and their
22 preferences so that when the user next logs on, the
23 system can skip the steps required to obtain that
24 data for the new session. The use of cookies within
25 the broker-to-broker system can be severely
26 restricted if not eliminated so as to minimize any
27 security risk, e.g. items like colors, and language
28 preference, frame sizes and positions could most
29 probably be safely stored in cookies, but items such
30 as security identifiers, default fulfilling members
31 and default execution venue must not be stored since

1 knowledge of these could be used to interpret the
2 fulfilling brokers transaction history.

3 BIWEB and SELWEB may be constructed so that its
4 operating "footprint" on the user's machine is as
5 lightweight or small as possible. It must not be
6 considered the norm that users will be technically
7 literate and that they will have high-powered
8 machines. One target machine environment may be a
9 Pentium 233 MHz personal computer running the
10 Microsoft Windows NT operating system.

11 Because BIWEB and SELWEB will be used at many
12 sites, and very often where the level of technical
13 support is minimal or self-help is the norm, BIWEB
14 and SELWEB must not require the installation of any
15 components from disk/CDROM etc. at the customer's
16 premises. For security purposes every fulfilling
17 member will need to use a physical security token
18 unique to each individual at the fulfilling member's
19 premises, such as SecurID from RSA Dynamics, to
20 identify themselves to the system. This token(s) and
21 operating guide will be delivered to the fulfilling
22 member using secure courier facilities. Once in
23 possession of the token, all that will be required of
24 the user will be to start up their web browser and
25 navigate to the BIWEB or SELWEB URL. If there are
26 software components to be installed BIWEB or SELWEB
27 will automatically detect that condition and carry
28 out the necessary operations.

29 The access network (referred to herein as BICON)
30 is the physical infrastructure that connects the
31 originating members 1230 with the broker-to-broker

1 processing system 200. In the above embodiment, BICON
2 comprises secure digital communications links between
3 individual members and the broker-to-broker
4 processing system 200. In one modification, BICON is
5 a Virtual Private Digital Network, VPDN which is a
6 physically secure, and information secure, guaranteed
7 quality service, QOS, data communications network.
8 Such networks can be installed between the
9 originating member's location and the nearest broker-
10 to-broker point-of-presence. The contract for
11 installation and operation is between the originating
12 member and broker-to-broker systems. In another
13 modification, BICON comprises the internet.

14 Access to BICON may be controlled through the
15 use of firewall 1210, 1215 and RSA Dynamics SecurID
16 with Radius database technology 1200, 1205 which
17 provides physical access token technology. In one
18 embodiment, every individual at each originating
19 member who may use BINET, has an individual logon
20 identity, and must be in possession of his/her unique
21 SecureID token. The use of generic IDs or broker-wide
22 IDs may be contractually prohibited by a broker-to-
23 broker network operator in its agreements with
24 customers.

25 All data packets sent over BICON may be
26 encrypted using 128 bit SSL technology so as to
27 ensure complete privacy and confidentiality of the
28 information.

29 BICON is available in several different physical
30 implementations depending on the volume and response
31 characteristics required by the originating member.

1 In addition if and when the broker-to-broker system's
2 customers use the value-added features of the broker-
3 to-broker system service, e.g. interactive voice
4 messaging, they will need to upgrade the
5 communications bandwidth to minimize adverse impact
6 to the transaction components. A major consideration
7 in the implementation of BICON is to minimize
8 propagation or latency delays.

9 For occasional or light usage, i.e. a single
10 BIWEB session a 56K V90 modem dial-up session should
11 suffice. Dial-up connections may be used almost
12 always to initially connect the broker-to-broker
13 systems customers into BINET, while higher capacity
14 is being installed. Dial-up connections will also be
15 of use to those originating members who have a field
16 force, mobile workers, and/or very small outlying
17 branch offices. In addition, originating members may
18 most probably use dial-up connections in the less
19 well developed countries where order volume will be
20 small, and the telecommunications infrastructure is
21 not developed and/or slow to respond.

22 For more demanding light usage, i.e. 2 BIWEB
23 sessions, a 56/64Kbit ISDN dial-up session should
24 suffice. ISDN or high bit rate digital interfaces for
25 dial-up connections may be seen as the preferable
26 alternative to analogue dial-up where the
27 telecommunications infrastructure allows and/or the
28 break-even point between dial-up and fixed line
29 economics has not been reached.

30 For those originating members who choose to
31 implement computer connections into BINET using

1 BIAPI, it is expected that different
2 characterizations will apply which will be specific
3 to that originating member's needs and expectations.

4 SELCON is the physical networking infrastructure
5 that connects the fulfilling members with the broker-
6 to-broker system processing core. In one modification
7 SELCON is a Virtual Private Digital Network, VPDN
8 1320. SELCON is a physically secure, and information
9 secure, guaranteed quality service, QOS, data
10 communications network.

11 SELCON is installed between the fulfilling
12 member's location and the nearest the broker-to-
13 broker system POP (point of presence). In another
14 modification SELCON comprises the internet.

15 Access to SELCON may be controlled through the
16 use of RSA Dynamics SecurID 1300, 1305 which provides
17 physical access token technology. Every individual at
18 each fulfilling member who will use SELNET, must have
19 an individual logon identity, and must be in
20 possession of his/her unique SecureID token. The use
21 of generic IDs or broker-wide IDs may be
22 contractually prohibited by the broker-to-broker
23 network operator in its agreements with the broker-
24 to-broker system's customers.

25 All data packets sent over SELCON may be
26 encrypted using 128 bit SSL technology so as to
27 ensure complete privacy and confidentiality of the
28 information.

29 SELCON is available in several different
30 physical implementations depending on the volume and
31 response characteristics required by the fulfilling

1 member. In addition if and when broker-to-broker's
2 customers use the value-added features of the broker-
3 to-broker service, e.g. interactive voice messaging,
4 they will need to upgrade the communications
5 bandwidth to minimize adverse impact to the
6 transaction components. A major consideration in the
7 implementation of SELCON is to minimize propagation
8 or latency delays.

9 For occasional or light usage, i.e. a single
10 SELWEB session, a 56K V90 modem dial-up session
11 should suffice. Dial-up connections may be used
12 almost always to initially connect the broker-to-
13 broker system 's customers into SELNET, whilst higher
14 capacity is being installed. In addition, fulfilling
15 members will most probably use dial-up connections in
16 the less well developed countries where order volume
17 will be small, and the telecommunications
18 infrastructure is not developed and/or slow to
19 respond.

20 For more demanding light usage, i.e. 2 SELWEB
21 sessions, a 56/64Kbit ISDN dial-up session should
22 suffice. ISDN or other high bit rate digital dial-up
23 connections will be seen as the preferable
24 alternative to analogue dial-up where the
25 telecommunications infrastructure allows and/or the
26 break-even point between dial-up and fixed line
27 economics has not been reached.

28 For higher volumes, improved response times, and
29 where on-line times cross the classic break-even
30 point between dial-up and fixed-line, fixed or leased
31 lines will need to be installed. The minimum fixed-

1 line speed available for the broker-to-broker system
2 may be 64Kbits/second. Higher speeds will need to be
3 installed in proportion to the number of concurrent
4 the broker-to-broker system sessions in use at the
5 fulfilling member's location, and the response times
6 which they wish to achieve. As a rule of thumb, 2
7 SELNET sessions can be concurrently handled per
8 64Kbit/sec segment when using just SELWEB. For those
9 fulfilling members who choose to implement computer
10 connections into SELNET using SELAPI, it is expected
11 that different characterizations will apply which
12 will be specific to that originating broker's needs
13 and expectations.

14

15 SUMMARY

16 Thus, preferred embodiments provide an order
17 delivery and management facility for use by members
18 to communicate with each other and to their
19 respective settlement agents. The system includes an
20 integrated network of applications through which a
21 member seeking fulfillment of a securities
22 transaction order may enter the order into the system
23 for routing to another designated member who is a
24 broker qualified to fulfill the order or part thereof
25 on the basis of prevailing conditions in the
26 fulfilling member's market, subject to conditions
27 specified by the originating member. Nothing
28 inherent in the system itself brings together orders
29 or trading interests. The parties using the system
30 may be limited to originating members, fulfilling
31 members and settlement agents. Brokers or for

1 example, investment banks may use the system as
2 originating or fulfilling members or both. Investors
3 need not have access to the system other than through
4 members.

5 Originating members using the system are charged
6 a flat annual fee for up to a specified number of
7 orders that result in the generation by the system of
8 settlement instructions upon execution of a trade,
9 based on an estimate of anticipated usage of the
10 system by each such broker. The annual fee may vary
11 according to a specific member's discount schedule or
12 system-wide transaction volume schedules. A
13 relatively nominal excess usage fee may be charged
14 only if the originating member's use of the system
15 generates a number of settlement instructions in
16 excess of the applicable annual fee limit, in order
17 to compensate the network operator for the additional
18 use of the system's transmission and processing
19 capabilities. Similarly, the network operator may
20 assess excess usage fees to fulfilling members who
21 generate more than a fixed number of settlement
22 instructions through the system on orders sent to
23 them through the system that they have executed.
24 Although excess usage fees are ultimately calculated
25 based on the total number of executed trades
26 processed to the settlement subsystem, as a means of
27 measuring usage of the system, the receipt of such
28 fees are not dependent on the completion of
29 transactions as no rebate are paid if a trade
30 submitted to settlement does not actually close
31 absent a failure of the system. If a transaction is

1 re-run through the settlement subsystem due to a
2 system failure or error, a credit may be accorded to
3 the party charged.

4 The system does not change the sequencing of
5 orders received from the originating member, nor
6 prioritizes orders from any given originating member,
7 nor among originating members. The system does not
8 change the sequencing of fulfillment advisories
9 transmitted from a fulfilling member back to an
10 originating member.

11 Agreements between the network operator and
12 members contracting to use the system may require
13 that brokers have and maintain registration or
14 qualification as broker-dealers to the extent
15 required in the respective jurisdictions in which
16 they conduct business, including the registration of
17 foreign brokers whose activities in the United States
18 or with U.S. investors require registration as
19 broker-dealers under the Exchange Act. Brokers using
20 the system may receive no warranty or guarantee from
21 the network operator concerning execution quality in
22 the fulfillment of orders.

23 In certain modified embodiments it is expected
24 that OMIF 202 may apply a significant amount of
25 sensibility checking to the order, which in the limit
26 may be as rigorous as the checks applied by broker-
27 to-broker processing system 200, but the intent of
28 the OMIF checks are mainly to provide the order
29 originator with a more rapid response time.

30 It will be apparent that certain alternative
31 embodiments of the broker-to-broker retain many of

1 the advantages of the preferred embodiments.
2 Specifically, a preferred broker-to-broker system
3 enables originating members to send orders for
4 execution to fulfilling members using a secure
5 connection. The resulting transactions can be cleared
6 and settled in the name of the two parties using the
7 broker-to-broker system who are able to manage the
8 process from end terminals.

9 The broker-to-broker system enables originating
10 members to enter orders for transmission to the
11 selected fulfilling members.

12 The broker-to-broker system enables fulfilling
13 members to receive order requests from the
14 originating members.

15 The broker-to-broker system enables fulfilling
16 members to enter fill/execution reports for
17 transmission back to the originating member.

18 The broker-to-broker system enables the
19 originating member to receive fill/execution
20 advisories/reports.

21 The broker-to-broker system enables either party
22 to review the status of their orders.

23 The broker-to-broker system enables either party
24 to request to modify attributes of the order.

25 The broker-to-broker system enables the
26 originating member and fulfilling member to receive
27 modification requests and to accept or reject the
28 request.

29 The broker-to-broker system enables either party
30 to request to cancel an order.

1 The broker-to-broker system enables either party
2 to receive cancellation requests and to accept or
3 reject the request.

4 The broker-to-broker system enables either party
5 to sign off an order, i.e. initiate the settlement
6 process for the fills/executions against his/her
7 order.

8 The broker-to-broker system enables the OM and
9 FM to receive sign off requests and to accept or
10 reject the request.

11 The broker-to-broker system provides settlement
12 management capability for those broker-to-broker
13 system customers who wish to contract out that
14 service to the broker-to-broker system. In this
15 instance, broker-to-broker system will generate and
16 control all messaging to and from the customer's
17 elected agent bank and custodian in the name of
18 broker-to-broker system customer.

19 The broker-to-broker system enables either party
20 to upload material, e.g. notes, pricing
21 models/tools/guidelines, audio clips, video clips to
22 central electronic information library.

23 The broker-to-broker system enables either party
24 to search the electronic information library and
25 download material.

26 The broker-to-broker system enables either party
27 to interrogate their billing account with broker-to-
28 broker system to obtain information such as:
29 transaction fees outstanding to broker-to-broker
30 system; customer's membership fees outstanding;

1 rebates applicable; agreed fee structure; and average
2 TPD for week, month, quarter.

3 The broker-to-broker system enables either party
4 to send messages directly to other broker-to-broker
5 system customer(s) using the secure system network
6 facilities.

7 The broker-to-broker system enables either party
8 to obtain information about their use of broker-to-
9 broker's facilities such as: transaction history by:
10 time period, security, market, exchange, sector,
11 geographical region, execution quality statistics,
12 VWAP, security control, authorized names, and
13 passwords.

14 The broker-to-broker system enables either party
15 to search for security information held within the
16 databases of the broker-to-broker system.

17 The broker-to-broker system enables either party
18 to search for member information held within the
19 databases of the broker-to-broker system.

20 Having described in detail embodiments of the
21 present invention, the following rules, legalities
22 and business principles may be applied.

23

24 TYPE OF ENTITY

25 Preferred embodiments of the broker-to-broker
26 system do not function as a broker, bank, clearing
27 house, exchange, crossing network, auction room,
28 aggregator or other place where the change of
29 beneficial ownership of securities takes place.
30 Other embodiments may however include one or more of
31 the above or other functions. The network provides a

1 conduit through which orders can be transmitted for
2 execution; described embodiments of broker-to-broker
3 system do not provide the capability to match buyer
4 and seller within it.

5

6 BANK AND SECURITIES ACCOUNTS

7 Broker-to-broker systems preferably never take
8 ownership of stock, or monies in the transaction.
9 Embodiments of the broker-to-broker system thus need
10 not have securities account(s) and shall not settle
11 securities for its "own account." In order to
12 prevent any confusion or, aspersion of front running
13 or other infringement of securities regulations,
14 people closely associated with the embodiments
15 described herein, e.g. employees, their direct
16 families, consultants, must conduct their investment
17 activities using a recognized and regulated
18 brokerage, and must submit copies of their brokerage
19 transaction records, and must seek prior approval
20 from the broker-to-broker system Compliance officer
21 so that the broker-to-broker system can protect its
22 reputation, assure its customers that confidentiality
23 is being observed, and assure the regulators that the
24 broker-to-broker system has put in place governance
25 to prevent abuse. (i.e. although the broker-to-broker
26 system is not a regulated entity, it conducts and
27 operates its business under the same if not more
28 rigorous set of rules as implemented by leading
29 financial institutions).

30 In preferred embodiments, the broker-to-broker
31 system does have a bank account(s), which exist

1 purely for the purposes of paying for goods,
2 services, taxes and other expenses, and for receiving
3 payment for services rendered to its members and
4 other income.

5

6 ROLE IN TRANSACTIONS

7 A preferred broker-to-broker system does not
8 take any commissions, or spreads on transactions the
9 broker-to-broker system's only financial involvement
10 in a transaction is the levy of a specific fees see
11 section Transaction Fees below.

12

13 TRANSACTION FEES

14 Preferred broker-to-broker systems charge an
15 annual membership fee, with additional fixed fees for
16 orders routed through the system and surcharges for
17 settlement support. The originating members are
18 responsible for all additional fees and surcharges
19 other than those surcharges for mistakes made by the
20 fulfilling member.

21

TRANSACTION SEQUENCING

22 Preferred embodiments of the broker-to-broker
23 system do not change the sequencing of orders
24 received from the originating member, i.e. there is
25 no prioritization of orders from any given
26 originating member, nor is there any prioritization
27 of orders amongst originating members; ALL members
28 and their orders are treated equally.

29

30 Broker-to-broker systems preferably do not
change the sequencing of execution/fill advisories

1 transmitted from the fulfilling member back to the
2 originating member.

3 REGULATORY POSITION

4 Since preferred embodiments of the broker-to-
5 broker system seek always to be regulatory and
6 business neutral, they do not oblige members to carry
7 out any action which may offend or otherwise
8 contravene the rules of their regulatory bodies or in
9 any way influence the investment and execution
10 decisions of its members. In particular, broker-to-
11 broker system will not solicit orders or give
12 financial advice and members of the system have no
13 obligation to deal with any other member(s). To that
14 extent, members of the broker-to-broker system shall
15 have no warranty or guarantee from the broker-to-
16 broker system concerning execution quality or service
17 obtained by members from other members.

18

19 BEST EXECUTION

20 In most jurisdictions the fulfilling broker of a
21 securities exchange is under a regulatory obligation
22 to provide "Best Execution", i.e. deliver the best
23 price possible, to his/her customer taking into
24 account all the factors surrounding the order, and
25 the state of the market; specific factors commonly
26 important to determining how to "best execute" an
27 order are related to the size of the order compared
28 with the normal market trading size so as to minimize
29 market impact, the urgency with which the customer
30 wants the order filled, e.g. the customer may be

1 willing to pay a premium to the prevailing market
2 price if his/her order can be filled in its entirety
3 in one immediate action rather than having to have
4 the order worked over the day where he/she is at risk
5 that the price might rise/fall due to some exogenous
6 event. In some markets transparency of transactions
7 is limited, thus in order to enable the originating
8 members to exercise more control over the quality of
9 their executions, the broker-to-broker system
10 provides at least two mechanisms:

11 The originating member can designate the
12 execution venue which they require the fulfilling
13 member to use when fulfilling the order, e.g. on
14 exchange, off inventory, cross, best execution. In
15 most jurisdictions the fulfilling member is released
16 from his/her obligation to obtain "best execution"
17 when their customer specifically demands a particular
18 method of execution. The broker-to-broker system
19 order entry front-end allows the originating member
20 to specify and set the default execution venue for
21 each market and member.

22 Every fill reported back by the fulfilling
23 member is time-stamped and the best official bid and
24 offer as published at that time by that fulfilling
25 member's market authority, (where available), is
26 automatically appended to the fill advisory. This
27 information enables the originating members to
28 compare the execution quality, which they have
29 received from their fulfilling member vs. the pricing
30 in the market. (Ideally the price stamped on the fill
31 should take the best bid and offer available for the

1 number of securities in the fill, e.g. if the
2 official best bid and offer is for 1,000 shares, but
3 the fill is for 5,000 shares then the best bid and
4 offer are priced incorrectly. This capability to more
5 precisely price stamp the fills is available in the
6 more transparent, more developed, and usually
7 electronic markets which to some extent reduces the
8 value of the price stamp to the originating member
9 from a "best execution" perspective, since such
10 markets are typically also more rigorously
11 regulated.)

12 REGULATORY REPORTING

13 On request the broker-to-broker system may on
14 behalf of the parties to the transaction generate and
15 transmit reports to the regulators of each party to
16 the transaction. Such reports are designated in the
17 name of the entity on whose behalf the broker-to-
18 broker system has generated and transmitted the
19 message.

20

21 SELECTING THE FULFILLING BROKERS

22 Preferred embodiments of the broker-to-broker
23 system do not prioritize amongst fulfilling members.
24 All orders from the originating member should fully
25 designate the identity of the desired fulfilling
26 member. The broker-to-broker system provides
27 facilities to enable the originating member to select
28 amongst the universe applicable for a given
29 security/market for their desired fulfilling member
30 and then to designate the selected fulfilling member
31 as their default selection for all subsequent orders

1 for that security/market until changed by the
2 originating member. Preferred embodiments thus cannot
3 select the fulfilling member on behalf of the
4 originating member. The broker-to-broker system
5 described herein does not change the identity of the
6 originating member's selected fulfilling member on an
7 order. Where an originating member's selection of
8 fulfilling member is invalid, the originating member
9 is prompted to choose a valid fulfilling member
10 before the order are further processed by the broker-
11 to-broker system.

12

13 TRANSACTION STRUCTURE

14 The two parties to a transaction across the
15 broker-to-broker system network are the initiating
16 member (in this case the member who enters an order
17 into the system, i.e. the originating member), and
18 the fulfilling member (in this case the member who
19 fills the order, i.e. the fulfilling member). The
20 legality/contract of the transaction is as follows:

21 Let A be the originating member issuing a buy
22 stock

23 Let B be the fulfilling member fulfilling the
24 buy order

25 Then the transaction, which takes place, is:

26 B sells stock to A

27 and

28 A buys stock from B

29 Note At no time does this embodiment of the
30 broker-to-broker system enter into the transaction,
31 thus concepts such as riskless principal, agency,

1 principal are completely excluded from the broker-to-
2 broker system business, legal, and financial
3 processes.

4

5 CONFIDENTIALITY

6 Preferably, elements of the broker-to-broker
7 system do not know the identity of the entity/person
8 on whose behalf the originating member entered the
9 order. Any reference code attached to the order by
10 the originating member are preserved and attached to
11 all messages relating to that order and its
12 subsequent executions, however embodiments of the
13 broker-to-broker system need not translate/map that
14 reference code or in any other way alter such
15 reference code. It is possible that the originating
16 member may need to attach such reference code to
17 enable them within their systems to be able to
18 determine the eventual beneficiary of the proceeds of
19 the transactions. The broker-to-broker system need
20 not store or otherwise record such reference codes
21 and need not use the presence of or omission of such
22 reference codes for any purpose whatsoever.

23 In addition, preferred embodiments of the
24 broker-to-broker system do not know the identity of
25 the entity from/to which the fulfilling member
26 bought/sold or otherwise obtained the securities in
27 order to fulfil the originating member's order.

28

29 PREFERRED SETTLEMENT ARRANGEMENTS

30 Settlement of the trade between the originating
31 member and their customer is within the purview of

1 the originating member. For example, the price
2 charged to the originating member's customer, fees,
3 quantity of shares received/delivered, date of
4 delivery, where delivered are at the originating
5 member's discretion. The broker-to-broker system
6 need not generate settlement instructions,
7 confirmations or other forms of advisories, messages
8 to the originating member's customer, unless
9 requested to by the originating member.

10 Settlement of the trade between the fulfilling
11 member and the "street" is within the purview of the
12 fulfilling member. The broker-to-broker system can
13 generate settlement instructions, confirmations or
14 other forms of advisories, and messages to the
15 fulfilling member's "street" counterparties.

16 The broker-to-broker system may on request and
17 on behalf of either or both parties to the
18 transaction generate and transmit settlement
19 instructions, designated and in their name to their
20 respective agent banks and custodians. In this case,
21 the preferred broker-to-broker system issues the
22 following set of settlement instructions:

23 Let A be the originating member issuing a buy
24 order for shares of stock, Z, quantity of shares = Y

25 Let B be the fulfilling member receiving the
26 order from A

27 Let the price of the execution be X Euros

28 Let the commission charged by the fulfilling
29 member be C%

30 Let the local taxes and other regulatory charges
31 for the fulfilling member's jurisdiction be T%

1 Party B instructions

2 Party B will deliver Y shares of security Z to
3 Party A

4 Party B will receive monies from party A
5 calculated in general as follows

6 $(X * Y) + [(X * Y) * C\%] + [(X * Y) * T\%] = \text{TOTAL}$

7 CONSIDERATION

8 In related embodiments, jurisdictional rules and
9 conventions are taken into account in the calculation
10 method shown above to allow for the different ways in
11 which taxes and other fees are levied in the
12 different markets.

13 Party A Instructions

14 Party A will receive Y shares of security Z

15 Party A will deliver monies to party B

16 calculated in general as follows:

17 $(X * Y) + (X * Y) * C\% + (X * Y) * T\%$

18 In related embodiments, jurisdictional rules are
19 conventions are further catered for in the
20 calculation method shown above to allow for the
21 different ways in which taxes and other fees are
22 levied in the different markets.

23 Party A will deliver monies to the broker-to-
24 broker system calculated as follows

25 A fixed fee if the total number of transactions
26 processed by PARTY A exceeds the number of
27 transactor's budget in Party A's annual fee.

28 The fixed transaction fee is purely exemplary
29 and may be varied according to, for example, per
30 originating member discount schedules, per
31 originating member rate, the broker-to-broker system

1 system-wide per transaction volume related pricing
2 schedules, the broker-to-broker system system-wide
3 per transaction pricing schedules based on
4 geography/market but in general the fee payable will
5 in no way be calculated off or in any other way be
6 related to the value of the security transaction
7 itself.

8

9 ACKNOWLEDGEMENTS AND ADVISORIES

10 On request, the broker-to-broker system may on
11 behalf of either or both parties to the transaction
12 generate trade confirmations or advisories to either
13 or both parties to the transaction. Thus for example
14 on behalf of the fulfilling member embodiments of the
15 broker-to-broker system generates a telex confirming
16 to the originating member the details of the
17 transaction prior to settlement processing so as to
18 permit the originating member's settlement function
19 to confirm that the transaction is as expected. Such
20 acknowledgements / advisories are designated in the
21 name of the entity on whose behalf the broker-to-
22 broker system has generated and transmitted the
23 message. Such acknowledgements / advisories include
24 the text similar to the following: "The information
25 contained within this acknowledgement and the terms
26 and conditions applicable at the time of execution
27 were agreed solely between you and the fulfilling /
28 originating member named below as a result of one or
29 more interactions across the broker-to-broker
30 network. Any disagreement with the following should

1 be notified immediately to the fulfilling /
2 originating member".

3

4 PROPRIETARY TRADING

5 The broker-to-broker application need never know
6 the "investor" details. Therefore the use of the
7 broker-to-broker network to carry out proprietary/own
8 account trading activities by the originating member
9 for the benefit of the originating member itself have
10 no significance to the way the broker-to-broker
11 system processes or otherwise acts upon the order and
12 any subsequent transaction resulting from that order.

13

14 CANCELS/AMENDMENTS

15 At any time in the transaction process the
16 originating member may alter/cancel any part or all
17 of the order. Once an order has been accepted by the
18 designated fulfilling member, any
19 alterations/cancellations to that order become
20 requests to alter/cancel which must be accepted by
21 the designated fulfilling member. In which case any
22 securities purchased/sold by the fulfilling member at
23 the time the alteration/cancellation request was
24 received must be received/delivered to the
25 originating member, i.e. settlement takes place if
26 any executions have taken place prior to the
27 fulfilling member receiving and accepting the change
28 request. That is, partial fills are still
29 contractually binding.

30

31

1 ORDER PRE-REQUISITES

- 2 Before an order is accepted by preferred
3 embodiments of broker-to-broker processing system 200
4 for onward routing to the fulfilling member, the
5 following minimum set of conditions have to be
6 satisfied, i.e. present on the "order form":
- 7 a) The security identifier is valid
 - 8 b) The number of securities to purchase/sell has
9 been specified
 - 10 c) The direction of the order has been
11 specified, i.e. buy, and sell
 - 12 d) The price conditions on the order are valid
13 for that specific market and that specific security,
14 e.g. market, limit
 - 15 e) The duration and timing of the order are
16 valid for the specific security and market, e.g. good
17 till end of day
 - 18 f) The trade date for the transaction has been
19 suggested
 - 20 g) The settlement date for the transaction has
21 been suggested
 - 22 h) The designated fulfilling broker for the
23 order is valid for the security and market
 - 24 i) The commission rate to be charged by the FM
25 has been suggested
 - 26 j) The desired execution venue has been selected
27 by the originating member
 - 28 k) The originating member may select "best
29 execution" as the venue, in which case the fulfilling
30 member may use their discretion as to how to best
31 fulfil the order.

1 1) The identity of the issuing person at the
2 originating member has been positively authenticated
3 and is valid.

4

5

APPENDIX

6

7 The Appendix recites Java code used to implement the
8 preferred embodiment described herein with reference
9 to Figures 1 to 9.

APPENDIX

OrderManagerBean.java

```

// Description: This class manages the interface between both online users or
// electronic interfaces to the order management system, with regard to orders.
// The interface includes all actions that can be associated
// with an order.
//
//Package
package com.sapient.ordermanager;
//EJB session imports
import javax.ejb.SessionContext;
import javax.ejb.SessionBean;
// EJB naming and lookup imports
import javax.naming.InitialContext;
import javax.naming.Context;
import java.util.Hashtable;

//Exception Classes
import java.rmi.RemoteException;
import javax.ejb.CreateException;
import javax.ejb.FinderException;
import javax.naming.NamingException;
import com.sapient.exception.B2BException;
import com.sapient.exception.LoggedException;

//Property Files
import java.util.Properties;

// Helper classes
import com.sapient.helper.B2BConstants;
import com.sapient.helper.Self;
import com.sapient.helper.StatusReturn;

// Framework class
import com.sapient.framework.ejb.SessionBeanAdapter;
import com.sapient.framework.ejb.common.HomeFactory;
import com.sapient.framework.ejb.common.HomeFactoryException;
import com.sapient.framework.logging.Logger;
import com.sapient.framework.config.ConfiguratorFinder;
import com.sapient.framework.config.PropertyNotFoundException;

```



```

//Order bean
import com.sapient.order.Order;
import com.sapient.order.OrderHome;
import com.sapient.order.OrderPK;
import com.sapient.order.OrderRequestElements;
import com.sapient.order.OrderDetailElements;

import com.sapient.audit.AuditLogProxy;
import com.sapient.audit.Action;

//Order Interface
import com.sapient.interfaces.OrderGTEIF;

import java.util.Date; //Not needed... just used for testing!

/**
 * @author Darach Ó Braonáin
 * @author Sapient Ltd
 * @version 1.0
 * @stereotype SessionBean
 * @remoteInterface com.sapient.ordermanager.OrderManager
 * @homeInterface com.sapient.ordermanager.OrderManagerHome
 */
public class OrderManagerBean extends SessionBeanAdapter
{
    private transient SessionContext context;
    private Order remoteOrderBean = null;

    private static final String thisClass = "OrderManagerBean.";
    //private static final String attrInt = "0";
    /**
     * Sets the context of the bean
     * @param context The Bean's Context
     */
    public void setSessionContext(SessionContext context)
    {
        this.context = context;
    }
    /**

```

```

* This method is called when the container picks this session object
* and assigns it to a specific session object. Insert code here to
* acquire any additional resources that it needs when it is in the
* ready state.
*/
public void ejbActivate()
{
}

/**
 * This method is called when the container disassociates the bean
 * from the session object identity and puts the instance back into
 * the pool of available instances. Insert code to release any
 * resources that should not be held while the instance is in the
 * pool.
 */
public void ejbPassivate()
{
}

/**
 * This method is invoked when a client invokes the matching create()
 * on the home interface.
 */
public void ejbCreate()
{
}

/**
 * The container invokes this method in response to a client-invoked
 * remove request. The bean's representation is removed from the
 * container.
 */
public void ejbRemove()
{
}

/**
 * This method is used to enter a new order. Then it is called, it must be

```

```

* passed a StatusObject, a Self object and an OrderDetailElements object. When
* it is called, it will call the EnterOrder method on the OrderInterface. This
* will enter the order in the GTE and then return an order number. It will
* then call the SubmitOrder method on the orderInterface which will then call
* the same method on the GTE. If this is successful, this method will then
* create a new order on the OrderBean using the orderReference that has been
* returned.
*
* @param StatusReturn
*
* This object is used to pass
* status data from method to
* method to indicate success or
* failure of certain processes.
*
* @param Self
*
* This object contains the
* UserId and BrokerId of the
* person who has logged into the
* system.
*
* @param OrderDetailElements
*
* The OrderDetailElements will
* contain all the order data
* that has been entered by the
* user as well as enriched
* information from the GTE such
* as the order number.
*
* @exception none
* @return void
*/
public void createNewOrder(StatusReturn theStatusReturn, Self theSelf, OrderDetailElements
theOrderDetailElements) throws B2BException, RemoteException {
    final String thisMethod = thisClass + "createNewOrder: ";

    // Debug logging to ensure that methods are being called correctly
    Logger.log(Logger.ENTRY, thisMethod + "Entered by: " + theSelf);

    // Ensure the LastActionBrokerId is set
    theOrderDetailElements.setLastActionBrokerId(theSelf.getBrokerId());

    // enterOrder() returns the new order reference.
    try {
        (new AuditLogProxy()).logAction(theSelf.getBrokerId(), Action.CREATE_NEW_ORDER, null, (new
OrderGTEIF()).enterOrder(theOrderDetailElements));
    } catch (CreateException ex) {
        throw new LoggedException("CreateException caught: " + ex);
    }
}

```

```

}

/**
 * This method is used to withdraw a pending request. It firstly calls the
 * WithdrawRequest on the OrderInterface. If this call is successful, this
 * method will then access the the OrderBean and call the WithdrawRequest
 * method which will clear the request fields and recalculate the order status.
 */
*
* @param StatusReturn
*
*
*
*
* @param Self
*
*
*
*
*
*
*
* @param OrderRequestElements
*
*
*
*
* @exception none
* @return void
*/
public void withdrawPendingRequest(StatusReturn theStatusReturn, Self theSelf, OrderRequestElements
theOrderRequestElements) throws B2BException, RemoteException {
    final String thisMethod = thisClass + "withdrawPendingRequest: ";

    // Debug logging to ensure that methods are being entered correctly
    Logger.log(Logger.ENTRY, thisMethod + "Entered by: " + theSelf);

    String myOrderReferenceNumber = theOrderRequestElements.getOrderId();
    OrderPK myOrderReferencePK = new OrderPK(myOrderReferenceNumber);

    // Ensure the LastActionBrokerId is set
    theOrderRequestElements.setLastActionBrokerId(theSelf.getBrokerId());

    (new OrderGTEIF()).withdrawRequest(theOrderRequestElements);

    try {

```

```

        (new
AuditLogProxy()).logAction(theSelf.getBrokerId(),Action.WITHDRAW_PENDING_REQUEST,null,theOrderRequestElements.getOrd
erId());
    } catch (CreateException ex) {
        throw new LoggedException("CreateException caught: "+ex);
    }
}

/**
 * This method is used to accept an order. It calls the method AcceptOrder on
 * the orderInterface. If this is successful, it calls the acceptOrder method
 * on the OrderBean.
 *
 * @param StatusReturn
 *
 * This object is used to pass
 * status data from method to
 * method to indicate success or
 * failure of certain processes.
 * This object contains the
 * UserId and BrokerId of the
 * person who has logged into the
 * system.
 *
 * @param orderDetailElements
 *
 * The OrderDetailElements will
 * contain all the order data
 * that has been entered by the
 * user as well as enriched
 * information from the GTE such
 * as the order number.
 *
 * @exception RemoteException
 * @return void
 */
public void acceptOrder(StatusReturn theStatusReturn, Self theSelf, OrderDetailElements
theOrderDetailElements) throws B2BException, RemoteException {
    final String thisMethod = this.getClass() + "acceptOrder: ";
    Logger.log(Logger.ENTRY,thisMethod +"Entered by: "+theSelf);

    String myOrderReferenceNumber = theOrderDetailElements.getOrderId();
    OrderPK myorderReferencePK = new OrderPK(myOrderReferenceNumber);

    // Ensure the LastActionBrokerId is set
    theOrderDetailElements.setLastActionBrokerId(theSelf.getBrokerId());

```

```

(new OrderGTEIF()).acceptOrder(theOrderDetailElements);

try { (new
AuditLogProxy()).logAction(theSelf.getBrokerId(), Action.ACCEPT_ORDER, null, theOrderDetailElements.getOrderId());
} catch (CreateException ex) {
    throw new LoggedException("CreateException caught: "+ex);
}

}

/**
 * This method is used to reject an unaccepted order. Firstly, the method
 * RejectOrder is called on the orderInterface. If this call is successful, the
 * method RejectOrder is called on the OrderBean.
 *
 * @param myStatusReturn This object is used to pass
 * status data from method to
 * method to indicate success or
 * failure of certain processes.
 * @param mySelf This object contains the
 * UserId and BrokerId of the
 * person who has logged into the
 * system.
 * @param OrderDetailElement The OrderDetailElements will
 * contain all the order data
 * that has been entered by the
 * user as well as enriched
 * information from the GTE such
 * as the order number.
 *
 * @exception none
 * @return void
 */
public void rejectOrder(StatusReturn theStatusReturn, Self theSelf, OrderDetailElements
theOrderDetailElements) throws B2BException, RemoteException {
    final String thisMethod = thisClass + "rejectOrder: ";

    Logger.log(Logger.ENTRY, thisMethod + "Entered by: " + theSelf.toString());

    String myOrderReferenceNumber = theOrderDetailElements.getOrderId();
    OrderPK myOrderReferencePK = new OrderPK(myOrderReferenceNumber);

```

```

// Ensure the LastActionBrokerId is set
theOrderDetailElements.setLastActionBrokerId(theSelf.getBrokerId());

(new OrderGTEIF()).rejectOrder(theOrderDetailElements);

try {
    (new
AuditLogProxy()).logAction(theSelf.getBrokerId(),Action.REJECT_ORDER,null,theOrderDetailElements.getOrderId());
} catch (CreateException ex) {
    throw new LoggedException("CreateException caught: "+ex);
}

}

/**
 * This method is used to cancel an unaccepted order. The cancellation is sent
 * to the orderInterface. This method in turn passes this cancellation on to
 * the GTE. If this is successful, the method CancelOrder will be called on the
 * OrderBean.
 *
 * @param StatusReturn
 *
 * This object is used to pass
 * status data from method to
 * method to indicate success or
 * failure of certain processes.
 * This object contains the
 * UserId and BrokerId of the
 * person who has logged into the
 * system.
 *
 * @param OrderDetailElements
 *
 * The OrderDetailElements will
 * contain all the order data
 * that has been entered by the
 * user as well as enriched
 * information from the GTE such
 * as the order number.
 *
 * @exception none
 * @return void
 */
public void cancelOrder(StatusReturn theStatusReturn, Self theSelf, OrderDetailElements
theOrderDetailElements) throws B2BException, RemoteException {
    final String thisMethod = thisClass + "cancelOrder: ";

    logger.log(logger.ENTRY,thisMethod +"Entered by: "+theSelf);

```

```

String myOrderReferenceNumber = theOrderDetailElements.getOrderid();
OrderPK myOrderReferencePK = new OrderPK(myOrderReferenceNumber);

// Ensure the LastActionBrokerId is set
theOrderDetailElements.setLastActionBrokerId(theSelf.getBrokerId());

// Ignoring returned request reference
(new OrderGTEIF()).requestCancelUnacceptedOrder(theOrderDetailElements);

try {
    (new
AuditLogProxy()).logAction(theSelf.getBrokerId(), Action.CANCEL_ORDER, null, theOrderDetailElements.getOrderid());
} catch (CreateException ex) {
    throw new LoggedException("CreateException caught: "+ex);
}
}

/**
 * This method is used to request the cancellation of an accepted order. The
 * request is sent to the orderInterface. This method in turn passes this
 * cancellation request on to the GTE. It does this by calling
 * RequestCancelOrder(...) method. If this is successful, the method
 * requestCancel will be called on the OrderBean.
 */
@param StatusReturn
*
*
*
* @param Self
*
*
*
* @param OrderRequestElements
*
*
*
*
* @exception none
* @return void
*/
This object is used to pass
status data from method to
method to indicate success or
failure of certain processes.
This object contains the
UserId and BrokerId of the
person who has logged into the
system.
The OrderRequestElements will
contain all the order data
that has been entered by the
user.

```



```

    public void requestCancelOrder(StatusReturn theStatusReturn, Self theSelf, OrderRequestElements
    theOrderRequestElements) throws B2BException, RemoteException {
        final String thisMethod = thisClass + "requestCancelOrder: ";

        Logger.log(Logger.ENTRY, thisMethod + "Entered by: " + theSelf.toString());

        String myOrderReferenceNumber = theOrderRequestElements.getOrderId();
        OrderPK myOrderReferencePK = new OrderPK(myOrderReferenceNumber);

        // Ensure the LastActionBrokerId is set
        theOrderRequestElements.setLastActionBrokerId(theSelf.getBrokerId());

        // Ignoring returned request reference
        (new OrderGTEIF()).requestCancelOrder(theOrderRequestElements);

        try {
            (new
                AuditLogProxy()).logAction(theSelf.getBrokerId(), Action.REQUEST_CANCEL_ORDER, null, theOrderRequestElements.getOrderId
                ());
        } catch (CreateException ex) {
            throw new LoggedException("CreateException caught: " + ex);
        }
    }

    /**
     * This method is used to accept the cancellation of an accepted order. The
     * acceptance is sent to the orderInterface. This method in turn passes this
     * cancellation acceptance on to the GTE. It does this by calling
     * AcceptCancelOrder(...) method. If this is successful, the method
     * acceptCancel will be called on the OrderBean.
     *
     * @param StatusReturn
     *
     * This object is used to pass
     * status data from method to
     * method to indicate success or
     * failure of certain processes.
     *
     * @param Self
     *
     * This object contains the
     * UserId and BrokerId of the
     * person who has logged into the
     * system.
     *
     * @param OrderRequestElements
     *
     * The OrderRequestElements will
    
```

```

*
*
*
* @exception none
* @return void
*/
public void acceptRequestCancelOrder(StatusReturn theStatusReturn, Self theSelf, OrderRequestElements
theOrderRequestElements) throws B2BException, RemoteException {
    final String thisMethod = thisClass + "acceptRequestCancelOrder: ";
    Logger.log(Logger.ENTRY, thisMethod + "Entered by: " + theSelf.toString());

    String myOrderReferenceNumber = theOrderRequestElements.getOrderid();
    OrderPK myorderReferencePK = new OrderPK(myOrderReferenceNumber);
    (new OrderGTEIf()).acceptCancelOrder(theOrderRequestElements);

    // Ensure the LastActionBrokerId is set
    theOrderRequestElements.setLastActionBrokerId(theSelf.getBrokerId());

    try {
        (new AuditLogProxy()).logAction(theSelf.getBrokerId(), Action.ACCEPT_REQUEST_CANCEL_ORDER,
null, theOrderRequestElements.getOrderid());
    } catch (CreateException ex) {
        throw new LoggedException("CreateException caught: " + ex);
    }
}

/**
 * This method is used to reject the cancellation of an accepted order. The
 * rejection is sent to the orderInterface. This method in turn passes this
 * cancellation rejection on to the GTE. It does this by calling
 * RejectModifyCancelOrder(...) method. If this is successful, the method
 * rejectCancel will be called on the OrderBean.
 *
 * @param StatusReturn
 *
 * This object is used to pass
 * status data from method to
 * method to indicate success or
 * failure of certain processes.
 * This object contains the
 * UserId and BrokerId of the
 * person who has logged into the
 * system.
 *
 * @param Self
 *
 *
 *
 *
 */

```

```

* @param OrderRequestElements
*
*
*
*
* @exception none
* @return void
*/
public void rejectRequestCancelOrder(StatusReturn theStatusReturn, Self theSelf, OrderRequestElements
theOrderRequestElements) throws B2BException, RemoteException {
    final String thisMethod = thisClass + "rejectRequestCancelOrder: ";
    Logger.log(Logger.ENTRY, thisMethod + "Entered by: " + theSelf.toString());

    String myOrderReferenceNumber = theOrderRequestElements.getOrderid();
    OrderPK myOrderReferencePK = new OrderPK(myOrderReferenceNumber);

    // Ensure the LastActionBrokerId is set
    theOrderRequestElements.setLastActionBrokerId(theSelf.getBrokerId());

    (new OrderGTEIF()).rejectCancelOrder(theOrderRequestElements);

    try { (new
AuditLogProxy()).logAction(theSelf.getBrokerId(), Action.REJECT_REQUEST_CANCEL_ORDER, null, theOrderRequestElements.get
Orderid());
    } catch (CreateException ex) {
        throw new LoggedException("CreateException caught: " + ex);
    }
}

/**
 * This method is used to request a modification of an accepted order. The
 * request is sent to the orderInterface. This method in turn passes this
 * modification request on to the GTE. It does this by calling
 * RequestModifyOrder(...) method. If this is successful, the method
 * requestModify will be called on the OrderBean.
 *
 * @param StatusReturn
 *
 * This object is used to pass
 * status data from method to
 * method to indicate success or

```

```

* @param Self
*
* This object contains the
* UserId and BrokerId of the
* person who has logged into the
* system.
*
* The OrderRequestElements will
* contain all the order data
* that has been entered by the
* user.
*
* @exception none
* @return void
*/
public void requestModifyOrder(StatusReturn theStatusReturn, Self theSelf, OrderRequestElements
theOrderRequestElements) throws B2BException, RemoteException {
    final String thisMethod = thisClass + "requestModifyOrder: ";

    Logger.log(Logger.ENTRY, thisMethod + "Entered by: " + theSelf.toString());

    String myOrderReferenceNumber = theOrderRequestElements.getOrderId();
    OrderPK myOrderReferencePK = new OrderPK(myOrderReferenceNumber);

    // Ensure the LastActionBrokerId is set
    theOrderRequestElements.setLastActionBrokerId(theSelf.getBrokerId());

    // Ignoring returned request reference
    (new OrderGTEIF()).requestModifyOrder(theOrderRequestElements);

    try {
        (new
AuditLogProxy()).logAction(theSelf.getBrokerId(), Action.REQUEST_MODIFY_ORDER, null, theOrderRequestElements.getOrderId
());
    } catch (CreateException ex) {
        throw new LoggedException("CreateException caught: " + ex);
    }
}

/**
 * This method is used to accept the modification of an accepted order. The
 * acceptance is sent to the orderInterface. This method in turn passes this
 * cancellation acceptance on to the GTE. It does this by calling

```

```

* AcceptModifyOrder(...) method. If this is successful, the method
* acceptModify will be called on the OrderBean.
*
* @param StatusReturn
*
* This object is used to pass
* status data from method to
* method to indicate success or
* failure of certain processes.
* This object contains the
* UserId and BrokerId of the
* person who has logged into the
* system.
*
* @param OrderRequestElements
*
* The OrderRequestElements will
* contain all the order data
* that has been entered by the
* user.
*
* @exception none
* @return void
*/
public void acceptRequestModifyOrder(StatusReturn theStatusReturn, Self theSelf, OrderRequestElements
theOrderRequestElements) throws B2BException, RemoteException {
    final String thisMethod = thisClass + "acceptRequestModifyOrder: ";
    Logger.log(Logger.ENTRY, thisMethod + "Entered by: " + theSelf.toString());

    String myOrderReferenceNumber = theOrderRequestElements.getOrderid();
    OrderPK myorderReferencePK = new OrderPK(myOrderReferenceNumber);
    (new OrderGTEIF()).acceptModifyOrder(theOrderRequestElements);

    // Ensure the LastActionBrokerId is set
    theOrderRequestElements.setLastActionBrokerId(theSelf.getBrokerId());

    try {
        (new AuditLogProxy()).logAction(theSelf.getBrokerId(), Action.ACCEPT_REQUEST_MODIFY_ORDER,
null, theOrderRequestElements.getOrderid());
    } catch (CreateException ex) {
        throw new LoggedException("CreateException caught: " + ex);
    }
}

/**
* This method is used to reject the modification of an accepted order. The

```

```

* rejection is sent to the orderInterface. This method in turn passes this
* cancellation rejection on to the GRE. It does this by calling
* RejectModifyCancelOrder(...) method. If this is successful, the method
* RejectModify will be called on the OrderBean.
*
* @param StatusReturn
*
* This object is used to pass
* status data from method to
* method to indicate success or
* failure of certain processes.
* This object contains the
* UserId and BrokerId of the
* person who has logged into the
* system.
*
* @param OrderRequestElements
*
* The OrderRequestElements will
* contain all the order data
* that has been entered by the
* user.
*
* @exception none
* @return void
*/
public void rejectRequestModifyOrder(StatusReturn theStatusReturn, Self theSelf, OrderRequestElements
theOrderRequestElements) throws B2BException, RemoteException {
    final String thisMethod = thisClass + "rejectRequestModifyOrder: ";
    Logger.log(Logger.ENTRY, thisMethod + "Entered by: " + theSelf.toString());

    String myOrderReferenceNumber = theOrderRequestElements.getOrderId();
    OrderPK myOrderReferencePK = new OrderPK(myOrderReferenceNumber);

    // Ensure the LastActionBrokerId is set
    theOrderRequestElements.setLastActionBrokerId(theSelf.getBrokerId());

    (new OrderGTEIF()).rejectModifyOrder(theOrderRequestElements);

    try {
        (new
        AuditLogProxy()).logAction(theSelf.getBrokerId(), Action.REJECT_REQUEST_MODIFY_ORDER, null, theOrderRequestElements.get
        OrderId());
    } catch (CreateException ex) {
        throw new LoggedException("CreateException caught: " + ex);
    }
}

```

```

/**
 * This method is used to request sign off of an order. The request for signoff
 * is sent to the orderInterface. This method in turn passes this signoff
 * request on to the GTE. It does this by calling the RequestSignoff(...)
 * method. If this is successful, the method requestSignoffOrder will be called
 * on the OrderBean.
 *
 * @param StatusReturn
 *
 * This object is used to pass
 * status data from method to
 * method to indicate success or
 * failure of certain processes.
 *
 * @param Self
 *
 * This object contains the
 * UserId and BrokerId of the
 * person who has logged into the
 * system.
 *
 * @param OrderDetailElements
 *
 * The OrderDetailElements will
 * contain all the order data
 * that has been entered by the
 * user as well as enriched
 * information from the GTE such
 * as the order number.
 *
 * @exception none
 * @return void
 */
public void requestSignoffOrder(StatusReturn theStatusReturn, Self theSelf, OrderRequestElements
theOrderRequestElements) throws B2BException, RemoteException {
    final String thisMethod = thisClass + "requestSignoffOrder: ";
    Logger.log(Logger.ENTRY, thisMethod + "Entered by: " + theSelf.toString());
    String myOrderReferenceNumber = theOrderRequestElements.getOrderid();
    OrderPK myOrderReferencePK = new OrderPK(myOrderReferenceNumber);
    // Ensure the LastActionBrokerId is set
    theOrderRequestElements.setLastActionBrokerId(theSelf.getBrokerId());
    // Ignoring returned request reference
    (new OrderGTEIF()).requestSignoff(theOrderRequestElements);
    try {

```

```

        (new AuditLogProxy()).logAction(thisSelf.getBrokerId(), Action.REQUEST_SIGNOFF_ORDER,
null, theOrderRequestElements.getOrderId());
    } catch (CreateException ex) {
        throw new LoggedException("CreateException caught: "+ex);
    }
}

/**
 * This method is used to accept sign off of an order. The acceptance of
 * signoff is sent to the orderInterface. This method in turn passes this
 * signoff acceptance on to the GTE. It does this by calling the
 * AcceptSignoff(...) method. If this is successful, the method
 * AcceptSignoffOrder will be called on the OrderBean.
 *
 * @param StatusReturn
 *
 * This object is used to pass
 * status data from method to
 * method to indicate success or
 * failure of certain processes.
 * This object contains the
 * UserId and BrokerId of the
 * person who has logged into the
 * system.
 * The OrderDetailElements will
 * contain all the order data
 * that has been entered by the
 * user as well as enriched
 * information from the GTE such
 * as the order number.
 *
 * @param Self
 *
 * @param OrderDetailElements
 *
 * @exception none
 * @return void
 */
public void acceptRequestsSignoffOrder(StatusReturn theStatusReturn, Self theSelf, OrderRequestElements
theOrderRequestElements) throws B2BException, RemoteException {
    final String thisMethod = thisClass + "acceptSignoffOrder: ";
    Logger.log(Logger.ENTRY, thisMethod + "Entered by: "+theSelf.toString());

    String myOrderReferenceNumber = theOrderRequestElements.getOrderId();
    OrderPK myOrderReferencePK = new OrderPK(myOrderReferenceNumber);

```



```

// Ensure the LastActionBrokerId is set
theOrderRequestElements.setActionBrokerId(theSelf.getBrokerId());

(new OrderGTEIF()).acceptsSignoff(theOrderRequestElements);

try {
    (new AuditLogProxy()).logAction(theSelf.getBrokerId(), Action.ACCEPT_SIGNOFF_ORDER,
null, theOrderRequestElements.getOrderId());
} catch (CreateException ex) {
    throw new LoggedException("CreateException caught: "+ex);
}

}

/**
 * This method is used to reject sign off of an order. The rejection of
 * signoff is sent to the orderInterface. This method in turn passes this
 * signoff rejection on to the GTE. It does this by calling the
 * RejectSignoff(...) method. If this is successful, the method
 * RejectSignoffOrder will be called on the OrderBean.
 *
 * @param StatusReturn
 *
 * This object is used to pass
 * status data from method to
 * method to indicate success or
 * failure of certain processes.
 * This object contains the
 * UserId and BrokerId of the
 * person who has logged into the
 * system.
 *
 * The OrderDetailElements will
 * contain all the order data
 * that has been entered by the
 * user as well as enriched
 * information from the GTE such
 * as the order number.
 *
 * @exception none
 * @return void
 */
public void rejectRequestSignoffOrder(StatusReturn theStatusReturn, Self theSelf, OrderRequestElements
theOrderRequestElements) throws B2BException, RemoteException {
    final String thisMethod = this.getClass() + "rejectSignoffOrder: ";

```

```

    Logger.log(Logger.ENTRY, thisMethod + "Entered by: " + thisSelf.toString());

    String myOrderReferenceNumber = theOrderRequestElements.getOrderId();
    OrderPK myOrderReferencePK = new OrderPK(myOrderReferenceNumber);

    // Ensure the LastActionBrokerId is set
    theOrderRequestElements.setLastActionBrokerId(theSelf.getBrokerId());

    (new OrderGTEIF()).rejectSignOff(theOrderRequestElements);

    try {
        (new
AuditLogProxy()).logAction(theSelf.getBrokerId(), Action.REJECT_SIGNOFF_ORDER, null, theOrderRequestElements.getOrderId
());
    } catch (CreateException ex) {
        throw new LoggedException("CreateException caught: " + ex);
    }
}
/**
 * This method is used to get all order details. It does this by calling the
 * getOrderDetails method on the OrderBean. This method returns an
 * OrderDetailElements object that contains all the order details. The details
 * that are required can then be extracted from this.
 *
 * @param myStatusReturn
 *
 * This object is used to pass
 * status data from method to
 * method to indicate success or
 * failure of certain processes.
 * This object contains the
 * UserId and BrokerId of the
 * person who has logged into the
 * system.
 * The OrderDetailElements will
 * contain all the order data
 * that has been entered by the
 * user as well as enriched
 * information from the GTE such
 * as the order number.
 *
 * @exception none
 * @return OrderDetailElements
 *
 * This variable will contain all
 * the order details. The ones
 * that are required can then be

```

130

```

*
*/
    public OrderDetailElements getOrderDetails(Self mySelf, OrderDetailElements theOrderDetailElements) throws
        B2BException, RemoteException {
        final String thisMethod = thisClass + "getOrderDetails: ";

        String OrderReferenceNumber = theOrderDetailElements.getOrderid();//this line will be replaced by the
        above code

        OrderPK myOrderReferencePK = new OrderPK(OrderReferenceNumber);
        OrderDetailElements myOrderDetailElements = new OrderDetailElements();

        try{
            OrderHome orderHome = (OrderHome) HomeFactory.findHome("com.sapiant.order.OrderHome");
            remoteOrderBean = orderHome.findByPrimaryKey(myOrderReferencePK);
            myOrderDetailElements = remoteOrderBean.getOrderDetails();
        } catch (FinderException ex) {
            throw new LoggedException(thisMethod + "orderid : " + theOrderDetailElements.getOrderid() + "
            brokerId : " + mySelf.getBrokerId() + " shadowedBy : " + mySelf.getShadowedBy() + " Could not find order with PK:
            [" + myOrderReferencePK + "]", ex);
        } catch (HomeFactoryException ex) {
            throw new LoggedException(thisMethod + "orderid : " + theOrderDetailElements.getOrderid() + "
            brokerId : " + mySelf.getBrokerId() + " shadowedBy : " + mySelf.getShadowedBy() + " Unable to get home factory for
            OrderHome", ex);
        }

        Logger.log(Logger.DEBUG, thisMethod + "orderid : " + theOrderDetailElements.getOrderid() + " brokerId : "
        + mySelf.getBrokerId() + " shadowedBy : " + mySelf.getShadowedBy() + " " + thisMethod + ": has returned the following data:
        " + myOrderDetailElements.toString());

        String userBrokerage = mySelf.getBrokerageId();

        Logger.log(Logger.DEBUG, thisMethod + "orderid : " + theOrderDetailElements.getOrderid() + " brokerId : "
        + mySelf.getBrokerId() + " shadowedBy : " + mySelf.getShadowedBy() + " " + thisMethod + ": User from Brokerage: " +
        userBrokerage + " attempting to retrieve order with OB: " + myOrderDetailElements.getOriginatingBrokerageId() + " and
        with FB: " + myOrderDetailElements.getFulfillingBrokerageId());

        if ( !(userBrokerage.equals(myOrderDetailElements.getOriginatingBrokerageId()) ||
        (userBrokerage.equals(myOrderDetailElements.getFulfillingBrokerageId())) )
            throw new LoggedException(thisMethod + "orderid : " + theOrderDetailElements.getOrderid() + "
            brokerId : " + mySelf.getBrokerId() + " shadowedBy : " + mySelf.getShadowedBy() + " " + thisMethod + "Authentication
            exception: The Order: " + myOrderDetailElements.getOrderid() + " could not be displayed!");

```

return myOrderDetailElements;

}

}

ExecutionManagerBean.java

```

/**
 * Title:      Execution Manager Bean <p>
 * Copyright:  Copyright © 2000 Broker To Broker Networks. All rights reserved. <p>
 * Company:    Sapient <p>
 * @author    Alan Cummins
 * @version   1.0
 *
 * @description This class manages the interface between both online users or
 * electronic interfaces to the order management system, with regard to order
 * executions (fills). The interface includes all actions that can be associated
 * with an execution (fill).
 */
package com.sapient.executionmanager;

import java.sql.Timestamp;

import javax.ejb.SessionContext;
import javax.ejb.SessionBean;
import javax.ejb.FinderException;
import javax.ejb.CreateException;
import javax.naming.InitialContext;
import javax.naming.Context;
import javax.naming.NamingException;
import java.rmi.RemoteException;
import java.util.Enumeration;
import java.util.ArrayList;
import com.sapient.framework.sql.GeneralSQLException;
import com.sapient.framework.ejb.common.ConnectionManager;
import com.sapient.framework.ejb.common.ConnectionManagerException;
import com.sapient.framework.config.ConfiguratorFinder;
import com.sapient.framework.config.PropertyNotFoundException;

import java.sql.Connection;
import java.sql.SQLException;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import com.sapient.framework.util.Day;

import com.sapient.order.Order;
import com.sapient.order.OrderHome;

```

```

import com.sapient.order.OrderPK;
import com.sapient.order.OrderDetailElements;

import com.sapient.interfaces.OrderGTEIF;

import com.sapient.order.OrderExecutionElements;
import com.sapient.execution.ExecutionHome;
import com.sapient.execution.ExecutionPK;
import com.sapient.execution.Execution;

import com.sapient.framework.ejb.common.HomeFactory;
import com.sapient.framework.ejb.common.HomeFactoryException;
import com.sapient.framework.logging.Logger;
import com.sapient.helper.B2BConstants;
import com.sapient.helper.Self;
import com.sapient.exception.B2BException;
import com.sapient.exception.LoggedException;
import com.sapient.audit.AuditLogProxy;
import com.sapient.audit.Action;

import com.sapient.framework.ejb.SessionBeanAdapter;

// EJB naming and lookup imports
import java.util.Hashtable;
import java.util.Properties;
import java.util.Calendar;

/**
 * Contains all methods used for execution management of an order.
 * Specifically management of fill actions and display of execution information
 * The associated execution detail elements are contained within a data access object
 * and are associated in a list of executions against an order
 *
 * @stereotype SessionBean
 * @remoteInterface com.sapient.ordermanagement.ExecutionManager
 * @homeInterface com.sapient.ordermanagement.ExecutionManager
 */
public class ExecutionManagerBean extends SessionBeanAdapter
{

```

```

/**

```

```

* Used for logging purposes
*/
private static String thisClass = "ExecutionManagerBean.";

/**
 * This method is invoked when a client invokes the matching create()
 * on the home interface.
 */
public void ejbCreate()
{
}

/**
 * The container invokes this method in response to a client-invoked
 * remove request. The bean's representation is removed from the
 * container.
 */
public void ejbRemove()
{
}

/**
 * Enters a fill against the associated order.
 * The order interface is called with all the required information for an execution.
 * The associated order has an execution element set with all the user-entered details.
 * The callback interface will update this order execution with any enriched data at a later stage.
 * @param orderExecutionElement All details relevant for entering a fill
 * @exception B2BException
 */
public void enterFill(Self self, OrderExecutionElements anOrderExecutionElement) throws B2BException,
RemoteException {
    // Call the interface and retrieve the execution reference
    Logger.log(Logger.DEBUG, "OrderId : " + anOrderExecutionElement.getOrderId() + "
ExecutionManagerBean.enterFill() called with OrderExecutionElements: [" + anOrderExecutionElement + ""]);
    (new OrderGTEIF()).executionEntry(anOrderExecutionElement);
}

try {
    (new
AuditLogProxy()).logAction(self.getBrokerId(), Action.ENTER_EXECUTION, null, anOrderExecutionElement.getOrderId());
} catch (CreateException ex) {
    throw new LoggedException("CreateException caught: " + ex);
}

```

```

    )
    /**
     * Requests the cancellation of an execution associated with an order.
     * The order interface is called to request the cancellation.
     * The order and the associated execution element are also set with the requested
     * cancellation information.
     * @param anOrderId
     * @param anExecutionReference
     * @param aCancelReason
     * @param anException B2BException
     */
    public void requestCancelExecution(Self self, String anOrderId, String anExecutionReference, String
    aCancelReason) throws B2BException, RemoteException {
        // Get the details of the relevant execution
        OrderExecutionElements myOrderExecutionElement = getExecution(anExecutionReference);
        myOrderExecutionElement.setExecutionCancelReason(aCancelReason);

        Logger.log(Logger.DEBUG, "orderId : " + anOrderId + " ExecutionManagerBean.requestCancelExecution() called
        with OrderExecutionElements: [" + myOrderExecutionElement + "], CancelReason: [" + aCancelReason + "]);
        (new OrderGTEIF()).requestCancelExecution(myOrderExecutionElement);

        try { (new
        AuditLogProxy()).logAction(self.getBrokerId(), Action.REQUEST_CANCEL_EXECUTION, null, anOrderId);
        } catch (CreateException ex) {
            throw new LoggedException("CreateException caught: " + ex);
        }
    }

    /**
     * Gets the list of executions against an order.
     * This function is used to return an ArrayList of OrderExecutionElements
     * associated with an Order. Does not return Cancelled executions.
     *
     * @returns ArrayList An ArrayList of OrderExecutionElements.
     * @param orderId
     * @return ArrayList A list of all executions associated with an order
     * @exception B2BException
     */
    public ArrayList displayExecutions(String orderId) throws B2BException, RemoteException {
        try {

```



```

ArrayList array = new ArrayList();
ExecutionHome executionHome =
    (ExecutionHome)HomeFactory.findHome("com.sapient.execution.ExecutionHome");
Enumeration executions = executionHome.findByOrderId(orderId);

while (executions.hasMoreElements()) {
    Execution execution = (Execution)executions.nextElement();
    array.add(execution.getOrderExecutionElements());
}
return array;
} catch (HomeFactoryException ex) {
    throw new LoggedException("OrderId : " + orderId + " " + thisClass +
        ":HomeFactoryException:displayExecutions:" + ex);
} catch (FinderException ex) {
    throw new LoggedException("OrderId : " + orderId + " " + thisClass +
        ":FinderException:displayExecutions:" + ex);
}
}

/**
 * Reject the cancellation of a fill.
 * Calls the rejectCancelFill method on the orderInterface.
 * This in turn calls the RejectCancelExecution(...) method on the GTE.
 * If this is successful, the method rejectCancelFill is called on the OrderBean.
 */
* @param      anOrderId
* @param      anExecutionReference
* @param      aRejectionReason
* @exception  B2BException
* @return     boolean
*/
public void rejectCancelExecution(Self self, String anOrderId, String anExecutionReference, String
    aRejectionReason) throws B2BException, RemoteException {
    // Retrieve all the relevant information for that execution
    OrderExecutionElements myOrderExecutionElement = getExecution(anExecutionReference);
    myOrderExecutionElement.setRejectExecutionCancelReason(aRejectionReason);

    // We need to populate this field, and it doesn't come back in the ExecutionElements info.
    try {
        OrderHome orderHome = (OrderHome)HomeFactory.findHome("com.sapient.order.OrderHome");
        Order order = orderHome.findByPrimaryKey(new OrderPK(anOrderId));
        OrderDetailElements details = order.getOrderDetails();

```

```

myOrderExecutionElement.setRequestReference(details.getRequestReference());
// XXX: This is a hack! The GTE expects FID_BROKER_ID and FID_USER_ID to be
// the current actioning user, whereas we use the terms to mean the fulfilling broker.
// This does work correctly, but the fields should be named more accurately.
myOrderExecutionElement.setExecutingBrokerageId(details.getOriginatingBrokerageId());
myOrderExecutionElement.setExecutedBrokerId(details.getOriginatingBrokerId());

    } catch (HomeFactoryException ex) {
        throw new LoggedException("OrderId : " + anOrderId+ " "+thisClass +
            ".HomeFactoryException:rejectCancelExecution:" + ex);
    } catch (FinderException ex) {
        throw new LoggedException("OrderId : " + anOrderId+ " "+thisClass +
            ".FinderException:rejectCancelExecution:" + ex);
    }

    (new OrderGTEIF()).rejectCancelExecution(myOrderExecutionElement);

    try {
        (new AuditLogProxy()).logAction(self.getBrokerId(), Action.REJECT_CANCEL_EXECUTION, null, anOrderId);
    } catch (CreateException ex) {
        throw new LoggedException("CreateException caught: "+ex);
    }
}

/**
 * Accept the cancellation of a fill.
 * Calls the acceptCancelFill method on the orderInterface.
 * This in turn calls the AcceptCancelExecution(...) method on the GTE.
 * If this is successful, the method acceptCancelFill is called on the OrderBean.
 */
    @param      anOrderId
    @param      anExecutionReference
    @exception  B2BException
    public void acceptCancelExecution(self self, String anOrderId, String anExecutionReference) throws
        B2BException, RemoteException {
        // Find the execution against which we wish to accept the cancellation
        OrderExecutionElements myOrderExecutionElement = getExecution(anExecutionReference);

        // We need to populate this field, and it doesn't come back in the ExecutionElements info.
        try {

```

```

OrderHome orderHome = (OrderHome)HomeFactory.findHome("com.sapiant.order.OrderHome");
Order order = orderHome.findByPrimaryKey(new OrderPK(anOrderId));
OrderDetailElements details = order.getOrderDetails();

myOrderExecutionElement.setRequestReference(details.getRequestReference());
// XXX: This is a hack! The GTE expects FID_BROKER_ID and FID_USER_ID to be
// the current actioning user, whereas we use the terms to mean the fulfilling broker.
// This does work correctly, but the fields should be named more accurately.
myOrderExecutionElement.setExecutingBrokerageId(details.getOriginatingBrokerageId());
myOrderExecutionElement.setExecutedBrokerId(details.getOriginatingBrokerId());
    } catch (HomeFactoryException ex) {
        throw new LoggedException("OrderId : " + anOrderId+" "+thisClass +
            " :HomeFactoryException:acceptCancelExecution:" + ex);
    } catch (FinderException ex) {
        throw new LoggedException("OrderId : " + anOrderId+" "+thisClass +
            " :FinderException:acceptCancelExecution:" + ex);
    }
}

(new OrderGTEIF()).acceptCancelExecution(myOrderExecutionElement);

try {
    (new AuditLogProxy()).logAction(self.getBrokerId(), Action.ACCEPT_CANCEL_EXECUTION, null, anOrderId);
} catch (CreateException ex) {
    throw new LoggedException("CreateException caught: "+ex);
}

}

/**
 * Status of a request to Cancel a fill
 *
 * <BR><BR><B>Screen Documentation:</B><BR>    FB_STCF_XXX_01, OB_STCF_XXX_01
 *
 * @param    orderID
 * @param    anExecutionReference
 * @exception B2BException
 * @return    orderExecutionElements
 */
public OrderExecutionElements displayExecution(String anOrderId, String anExecutionReference) throws
B2BException, RemoteException {
    return getExecution(anExecutionReference);
}

```

```

/**
 * Retrieve the execution Element that is pending cancel if any
 * There should always be only fill cancel pending against an order
 * @param orderId The order against which there is a pending cancel of fill
 * @return OrderExecutionElement The execution against which there is
 *         a cancel fill request
 *
 * @exception B2BException
 * @exception RemoteException
 */
public OrderExecutionElements retrievePendingCancelFillExecution(String orderId) throws B2BException,
RemoteException {
    try {
        OrderHome orderHome = (OrderHome)HomeFactory.findHome("com.sapiant.order.OrderHome");
        Order order = orderHome.findByPrimaryKey(new OrderPK(orderId));

        // This execution reference is the ID of the execution currently being cancelled, if any.
        String cancelExecutionRef = order.getOrderDetails().getRq_cancelExecutionRef();
        Logger.log(Logger.DEBUG, "OrderId : " + orderId+" retrievePendingCancelFillExecution() called, it
        found: ["+cancelExecutionRef+"]");

        if (cancelExecutionRef != null) {
            // Their is a request to cancel an execution outstanding.
            // Get the execution, and return it:
            try {
                ExecutionHome executionHome =
                    (ExecutionHome)HomeFactory.findHome("com.sapiant.execution.ExecutionHome");
                Execution execution = executionHome.findByPrimaryKey(new
                ExecutionPK(cancelExecutionRef));
                return execution.getOrderExecutionElements();
            } catch (HomeFactoryException ex) {
                throw new LoggedException("OrderId : " + orderId+" "+"Couldn't get ExecutionHome:
                "+ex);
            } catch (FinderException ex) {
                throw new LoggedException("OrderId : " + orderId+" "+"FinderException: Couldn't find
                execution reference: ["+cancelExecutionRef+"]", exception: "+ex);
            }
        } else throw new LoggedException("OrderId : " + orderId+" "+"No execution with request to cancel
        fill set against it, for OrderId: ["+orderId+"]");
    } catch (HomeFactoryException ex) {
        throw new LoggedException("OrderId : " + orderId+" "+"Couldn't get OrderHome: "+ex);
    } catch (FinderException ex) {

```

```

        throw new LoggedException("OrderId : " + orderId+ " "+"FinderException: Couldn't find order id:
["+orderId+"]", exception: "+ex");
    }

    /**
     * Returns an OrderExecutionElements object for the given execution reference.
     * A SQL statement is created and executed and the details of the first record
     * are used to set the attributes of an OrderExecutionElements object which
     * is returned.
     * <p>
     * <b>The function does not currently check for multiple records with the execution
     * reference passed in.</b> - it assumes execution references are unique across all orders.<p>
     *
     * @param myExecutionReference The execution reference of the execution.
     *
     * @returns myExecutionElements OrderExecutionElements containing all the
     *         details for that execution.
     */
    public OrderExecutionElements getExecution(String myExecutionReference) throws B2BException {
        try {
            ExecutionHome executionHome =
                (ExecutionHome)HomeFactory.findHome("com.sapient.execution.ExecutionHome");
            Execution execution = executionHome.findByPrimaryKey(new ExecutionPK(myExecutionReference));
            return execution.getOrderExecutionElements();
        } catch (HomeFactoryException ex) {
            throw new LoggedException("HomeFactoryException:getExecution:" + ex);
        } catch (FinderException ex) {
            throw new LoggedException("ExecutionManager:getExecution(): No execution with the Execution
            Reference [" + myExecutionReference + "] found.");
        } catch (RemoteException ex) {
            throw new LoggedException("RemoteException:getExecution:" + ex);
        }
    }
}

```

1 CLAIMS

2

3 1. A method of operating a computer system to
4 facilitate transactions on an exchange, a transaction
5 being performed by a fulfilling member having access
6 to the exchange on behalf of an originating party not
7 having access to the exchange, the method comprising:

8 receiving at a first interface of a processing
9 system at least one information item of an electronic
10 transaction proposal from an originating party;

11 transmitting at least one information item
12 relating to the electronic transaction proposal from
13 a second interface of the processing system to a
14 fulfilling member;

15 generating at the processing system settlement
16 criteria to be accepted by the originating party and
17 the fulfilling member; and

18 receiving from each of the originating party and
19 the fulfilling member an indication of acceptance of
20 the settlement criteria generated by the processing
21 system.

22

23 2. A method as in claim 1, wherein the step of
24 receiving the at least one information item of the
25 electronic transaction proposal comprises receiving
26 an information item indicating what is to be
27 transacted.

28

29 3. A method as in claim 1 or 2 , wherein the step
30 of receiving the at least one information item of the
31 electronic transaction proposal comprises receiving

1 an information item identifying a designated
2 fulfilling member to perform the transaction.

3

4 4. A method as in any preceding claim, wherein the
5 processing system receives settlement information
6 from said originating party and said fulfilling
7 member.

8

9 5. A method as in claim 4, wherein the settlement
10 criteria generated by the processing system are based
11 on settlement information received from the
12 originating party and the fulfilling member.

13

14 6. A method as in any preceding claim, wherein the
15 settlement criteria generated by the processing
16 system are based on stored settlement information
17 accessible by the processing system.

18

19 7. A method as in any preceding claim, including
20 the step of generating at said processing system
21 settlement instructions on behalf of said originating
22 party and said fulfilling member.

23

24 8. A method as in claim 7, wherein settlement
25 instructions are generated responsive to said
26 indications of acceptance of the settlement criteria.

27

28 9. A method as in any preceding claim, wherein a
29 fulfilling member requests a modification to at least
30 part of an electronic transaction proposal from an
31 originating party.

1
2 10. A method as in any preceding claim, wherein the
3 step of receiving the at least one information item
4 of the electronic transaction proposal comprises
5 receiving information items relating to a proposed
6 transaction selected from one or more of the
7 following:

8 a transaction type indicator;
9 a quantity indicator;
10 a price condition; and
11 timing information.
12

13 11. A method as in claim 10, wherein the timing
14 information indicates a proposed transaction date
15 and/or a proposed settlement date.
16

17 12. A method as in any preceding claim, wherein the
18 step of receiving the at least one information item
19 of the electronic transaction proposal comprises
20 receiving information items relating to the proposed
21 transaction from the originating party in two or more
22 sub-steps.
23

24 13. A method as in claim 12, wherein an originating
25 party provides a first information item in a first
26 sub-step and the processor system supplies a further
27 relevant information item responsive thereto.
28

29 14. A method as in claim 13, wherein the processing
30 system supplies a plurality of further relevant

1 information items from which the originating party
2 can select.

3

4 15. A method as in claim 14, wherein an originating
5 party provides a first information item indicating
6 what is to be transacted in a first sub-step and the
7 processing system prompts the originating party to
8 select from further information items representing
9 exchanges in a further sub-step.

10

11 16. A method as in any preceding claim, wherein the
12 processing system converts at least a portion of an
13 electronic transaction proposal from a first format
14 appropriate to the originating party into a second
15 format appropriate to the fulfilling member.

16

17 17. A method as in claim 16, wherein the processing
18 system converts a transactional term in a first
19 language appropriate to the originating party into a
20 second language appropriate to the fulfilling member.

21

22 18. A method as in claim 16 or 17, wherein the
23 processing system converts a price indication in a
24 first currency appropriate to the originating party
25 into a corresponding price indication in a second
26 currency appropriate to the fulfilling member.

27

28 19. A method as in claim 16, 17 or 18, wherein the
29 exchange is a securities exchange and the processing
30 system converts a first security identifier
31 appropriate to the originating party into a second

1 security identifier appropriate to the fulfilling
2 member.

3

4 20. A method as in claim 16, wherein the first
5 format is selected by the originating party.

6

7 21. A method as in claim 16, wherein the second
8 format is selected by the fulfilling member.

9

10 22. A method as in any preceding claim, wherein the
11 processing system transfers an information item
12 between the first interface and the second interface
13 without altering the format of the information item.

14

15 23. A method a in any preceding claim, wherein the
16 processing system generates a predetermined signal to
17 alert a member of a change in status of a proposed
18 transaction.

19

20 24. A method according to any preceding claim,
21 wherein a fulfilling member also functions as an
22 originating party by submitting a transaction
23 proposal for execution on a further exchange.

24

25 25. A method according to any preceding claim,
26 wherein an originating party also functions as a
27 fulfilling member by performing a transaction on a
28 further exchange.

29

30 26. A computer system for facilitating transactions
31 on an exchange, a transaction being performed by a

1 fulfilling member having access to the exchange on
2 behalf of an originating party not having access to
3 the exchange, the system comprising:

4 a first processing system interface adapted to
5 receive one or more information items of an
6 electronic transaction proposal from an originating
7 party;

8 a second processing system interface adapted to
9 transmit one or more information items relating to
10 the electronic transaction proposal to a fulfilling
11 member;

12 a processing system for routing communications
13 including information items to and from the first and
14 second interfaces, the processing system being
15 operable to generate settlement criteria to be
16 accepted by the originating party and the fulfilling
17 member, and wherein the processing system is arranged
18 to receive first and second information items
19 indicating acceptance of the settlement criteria by
20 each of the originating party and the fulfilling
21 member.

22

23 27. A computer system as in claim 26 wherein the
24 processing system comprises at least one further
25 processing system interface adapted to issue
26 settlement instructions in respect of a transaction
27 performed by the fulfilling member.

28

29 28. A computer system according to claim 26 or 27,
30 comprising a data store for holding settlement
31 information.

1
2 29. A computer system as in any of claims 26 to 28,
3 wherein the processing system is operable to generate
4 first and second settlement instructions in formats
5 appropriate to settlement agents of the originating
6 party and the fulfilling member.
7

8 30. A computer system as in claim 29, comprising a
9 data store for holding said first and second
10 information items indicating acceptance of the
11 settlement criteria by each of the originating party
12 and the fulfilling member.
13

14 31. A computer system as in claim 30, wherein the
15 processing system is operable to detect if the
16 originating party and the fulfilling member have
17 accepted the settlement criteria and to generate
18 settlement instructions in response thereto.
19

20 32. A computer system according to any of claims 26
21 to 31, wherein the processing system is operable to
22 generate prompts responsive to receiving an
23 incomplete electronic transaction proposal via said
24 first interface.
25

26 33. A computer system according to any of claims 26
27 to 32, wherein the processing system has access to a
28 data store holding data relevant to transactions and
29 is operable to access the data store in order to
30 provide one or more data items relevant to an
31 electronic transaction proposal.

1

2 34. A computer system as in any of claims 26 to 33,
3 comprising a data store for holding conversion
4 information for use in converting at least a portion
5 of an electronic transaction proposal from a first
6 format appropriate to an originating party into a
7 second format appropriate to a fulfilling member.

8

9 35. A computer system as in claim 34, wherein the
10 data store holds conversion information selected from
11 one or more of the following:

12 language conversion information;
13 currency conversion information; and
14 conversion information for transaction item
15 identifiers.

16

17 36. A computer system as in claim 35, wherein the
18 data store holds conversion information for
19 transaction item identifiers relating to securities.

20

21 37. A computer readable medium having stored therein
22 a set of general purpose routines for facilitating
23 transactions on exchanges, the computer readable
24 medium comprising:

25 a first routine for receiving at least one
26 information item of a transaction proposal from an
27 originating party;

28 a second routine for transmitting at least one
29 information item of the transaction proposal to a
30 fulfilling member;

1 a third routine for generating first and second
2 settlement criteria to be accepted by said
3 originating party and said fulfilling member; and
4 a fourth routine for receiving from each of the
5 originating party and the fulfilling member an
6 indication of acceptance of the settlement criteria.

7
8 38. Computer program code for facilitating
9 transactions on an exchange, a transaction being
10 performed by a fulfilling member having access to an
11 exchange on behalf of an originating party not having
12 access to the exchange, the program code comprising:

13 a first set of instructions for receiving at
14 least one information item of a transaction proposal
15 from an originating party;

16 a second set of instructions for transmitting at
17 least one information item of the transaction
18 proposal to a fulfilling member;

19 a third set of instructions for generating first
20 and second sets of settlement criteria to be accepted
21 by said originating party and said fulfilling member;
22 and

23 a fourth set of instructions for receiving from
24 each of said originating party and said fulfilling
25 member an indication of acceptance of the settlement
26 criteria.

27

28 39. A method of facilitating transactions on an
29 exchange, a transaction being performed by a
30 fulfilling member having access to the exchange on

1 behalf of an originating party not having access to
2 the exchange, the method comprising:
3 receiving at least one information item of a
4 transaction proposal from an originating party;
5 transmitting at least one information item of
6 the transaction proposal to a fulfilling member;
7 generating settlement criteria to be accepted by
8 the originating party and the fulfilling member; and
9 receiving indications of acceptance of the settlement
10 criteria from each of said originating party and said
11 fulfilling member.

12

13 40. A method of operating a computer system to
14 facilitate transactions on an exchange, a transaction
15 being performed by a fulfilling member having access
16 to the exchange on behalf of an originating party not
17 having access to the exchange, the method comprising:
18 receiving at a first interface of a processing
19 system an information item of an electronic
20 transaction proposal from an originating party, said
21 information item being in a first form appropriate
22 for the originating party;

23 converting said information item of the
24 electronic transaction proposal from the first form
25 into a second form appropriate to a fulfilling
26 member; and

27 transmitting the information item in the second
28 form from a second interface of the processing system
29 to a fulfilling member.

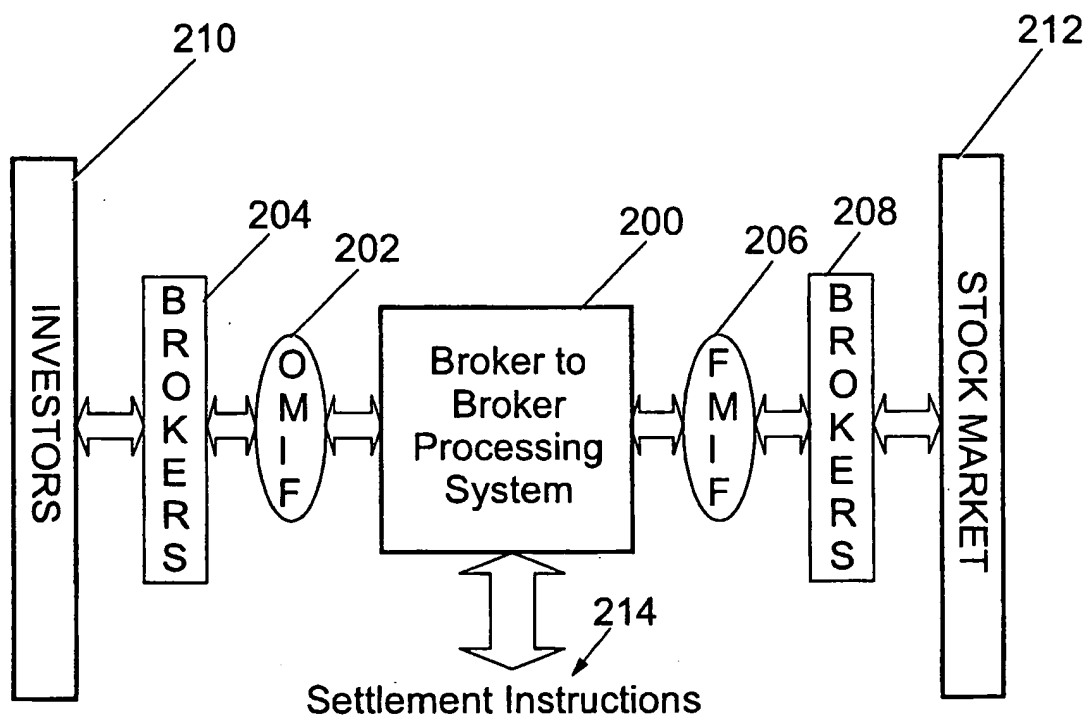
30

31 41. A computer system for facilitating transactions

1 on an exchange, a transaction being performed by a
2 fulfilling member having access to the exchange on
3 behalf of an originating party not having access to
4 the exchange, the system comprising:
5 a first processing system interface adapted to
6 receive an information item of an electronic
7 transaction proposal from an originating party, said
8 information item being in a first form appropriate to
9 the originating party;
10 a processing system operable to convert the
11 information item of the electronic transaction
12 proposal from the first form into a second form
13 appropriate to a fulfilling member; and
14 a second processing system interface adapted to
15 transmit the information item in the second form to
16 the fulfilling member.

17

1 / 13

*Fig. 1*

2 / 13

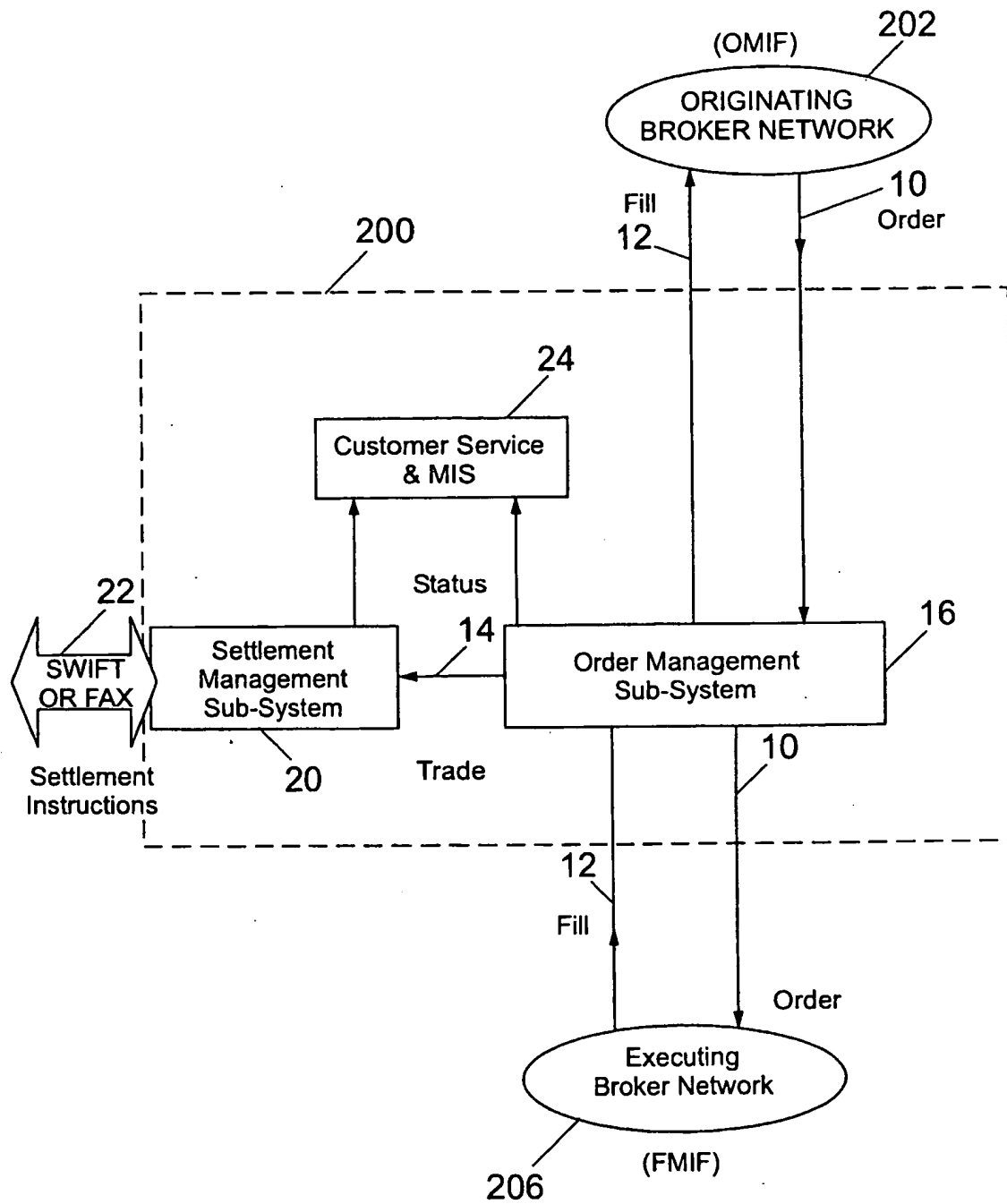


Fig. 2

3 / 13

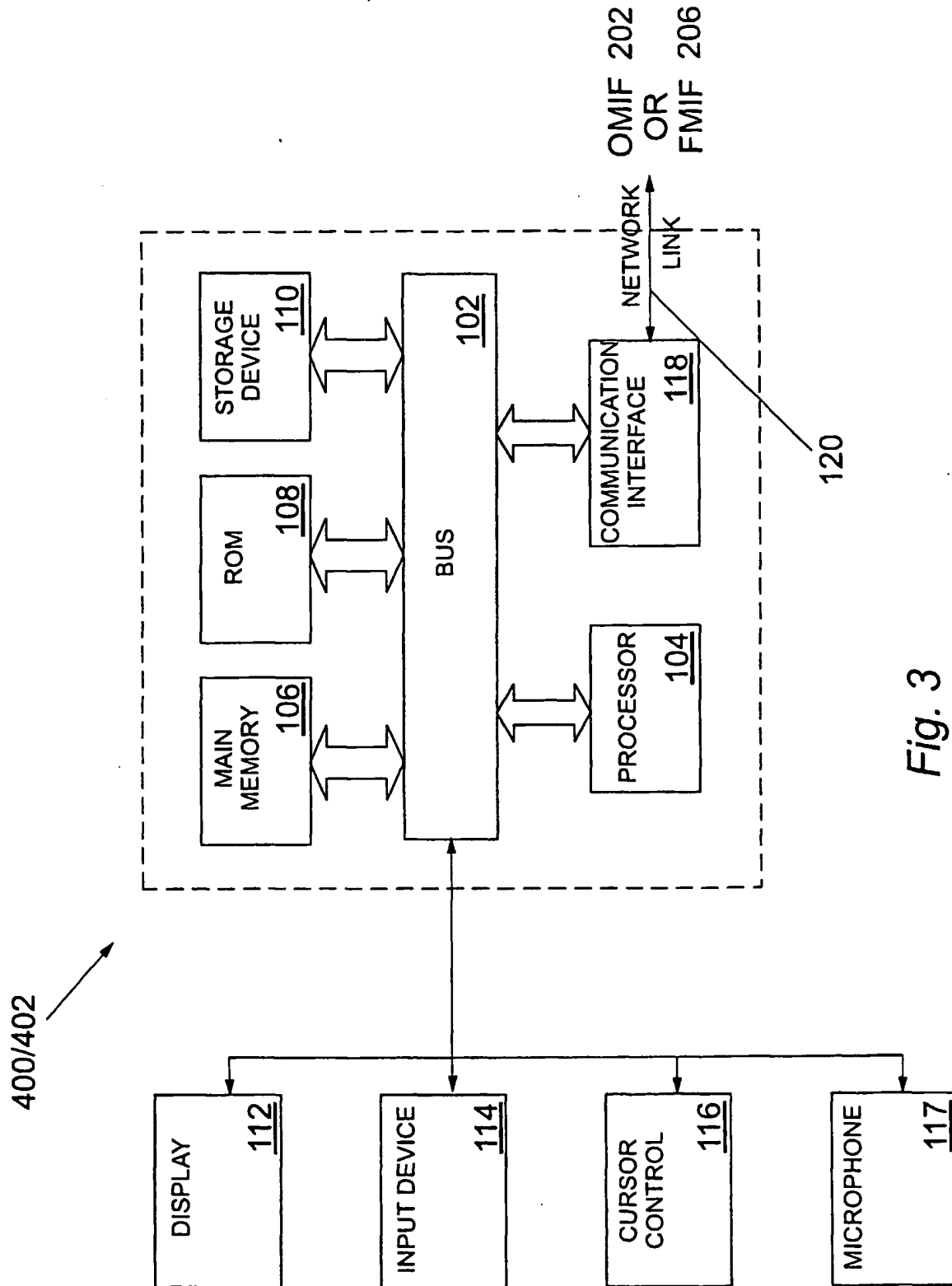


Fig. 3

4 / 13

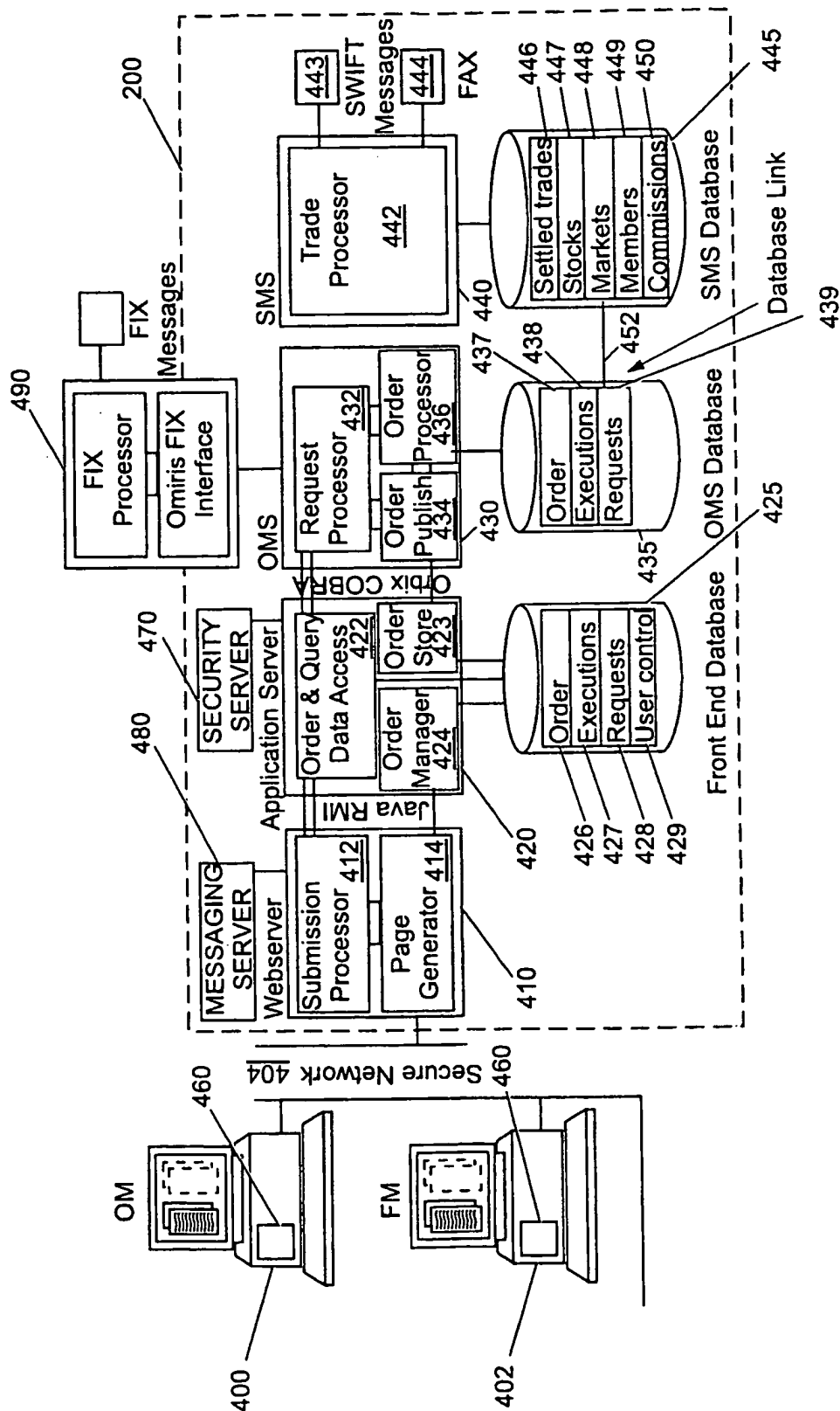
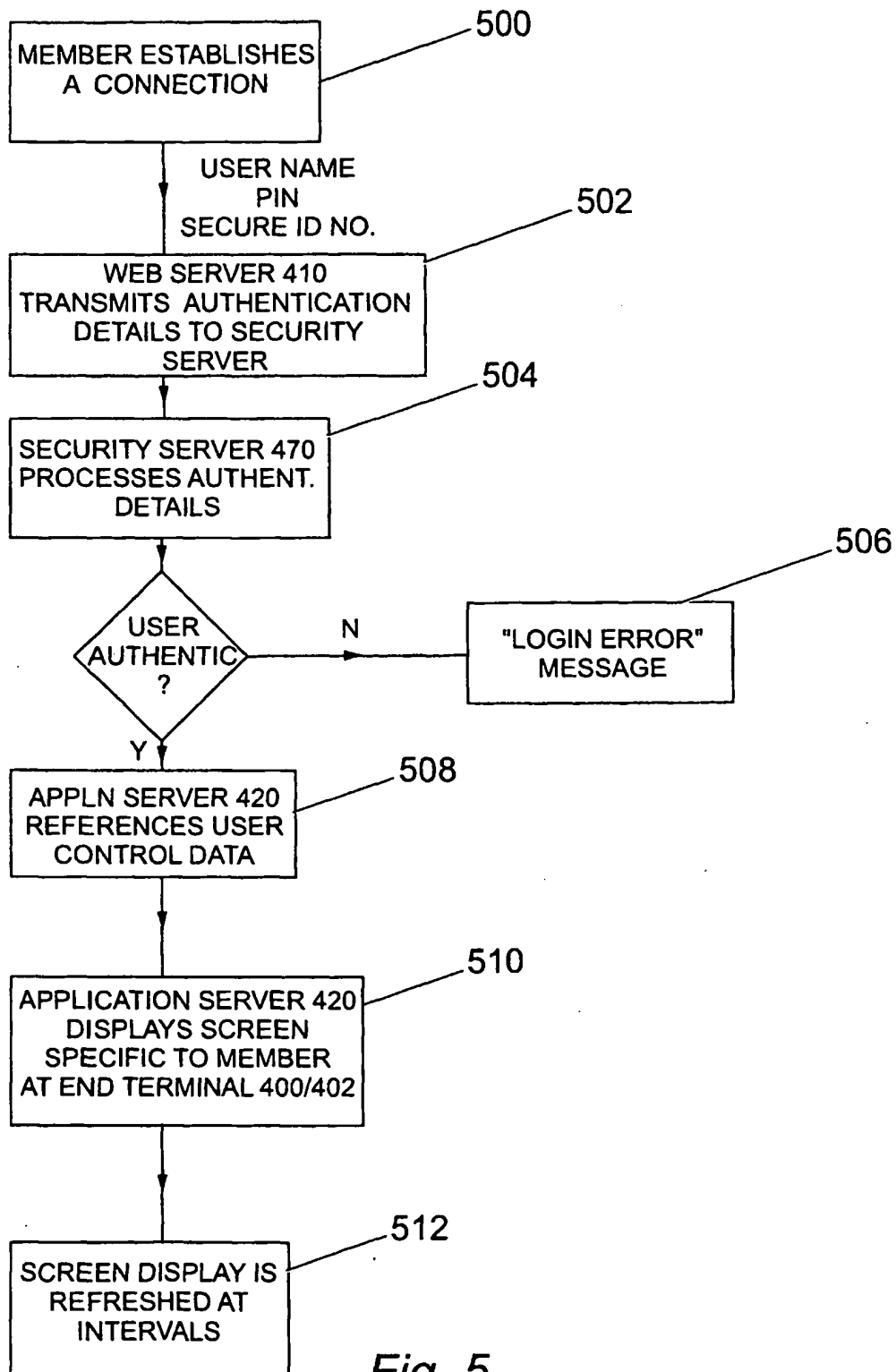


Fig. 4

5 / 13

*Fig. 5*

6 / 13

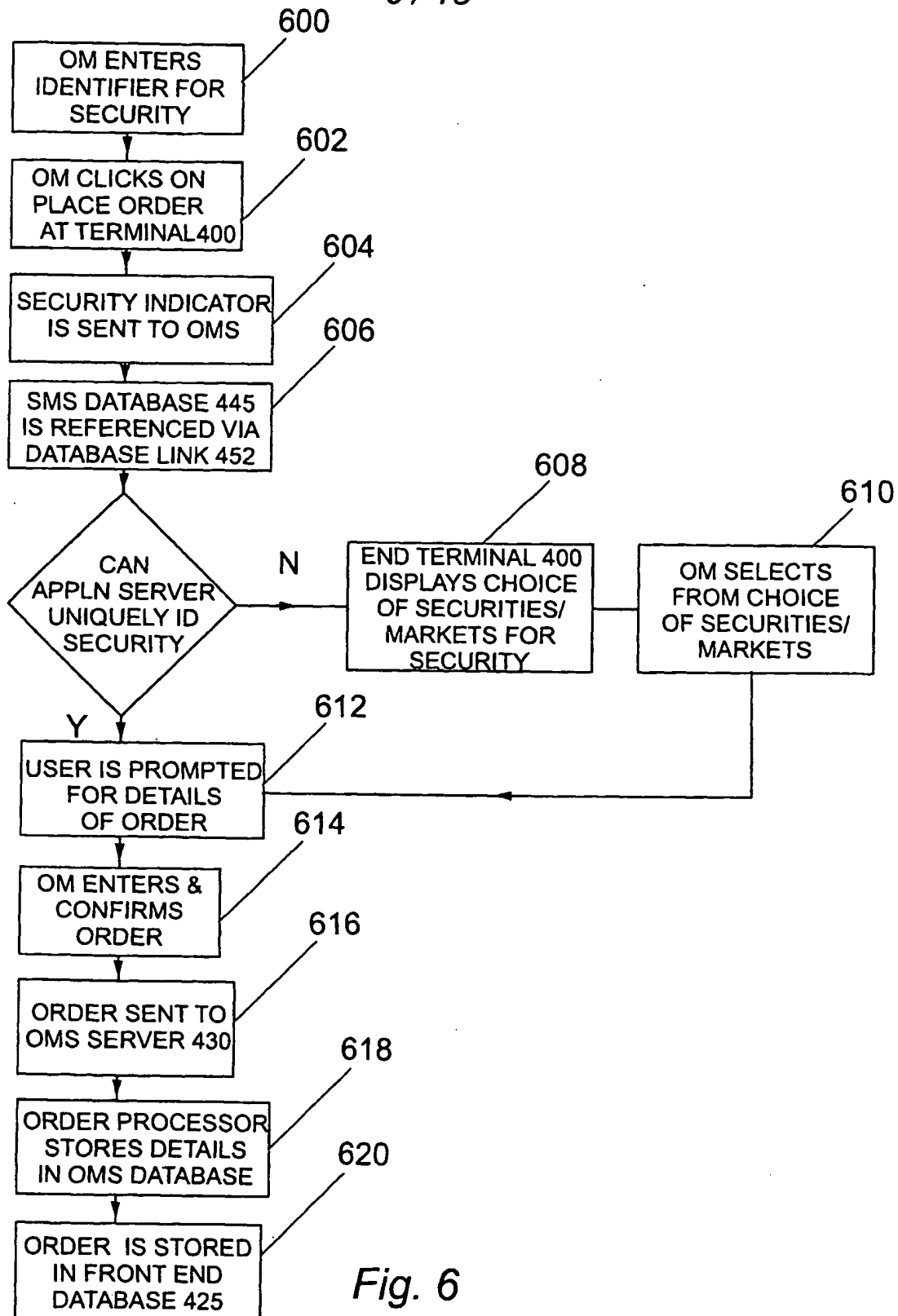


Fig. 6

7 / 13

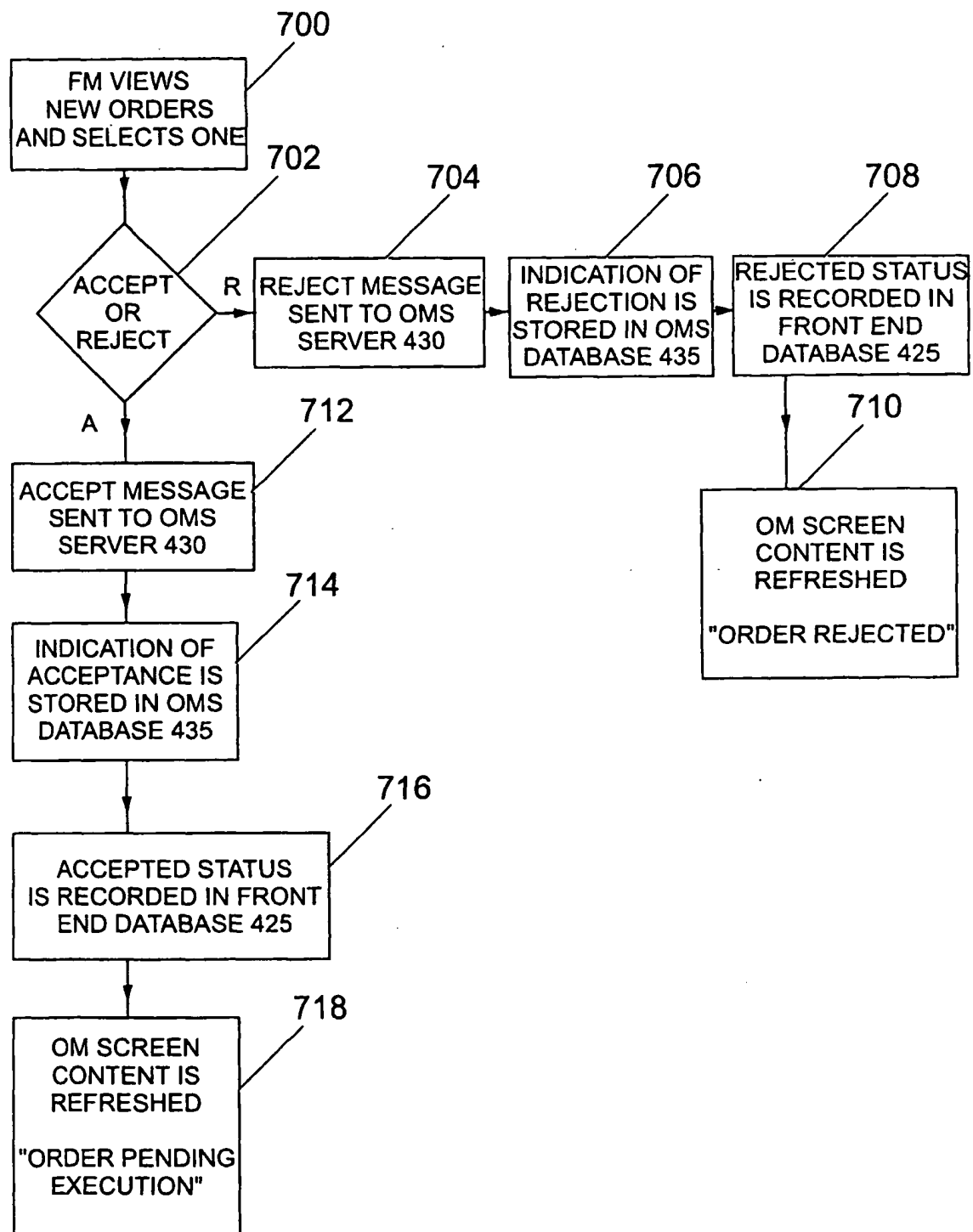


Fig. 7

8 / 13

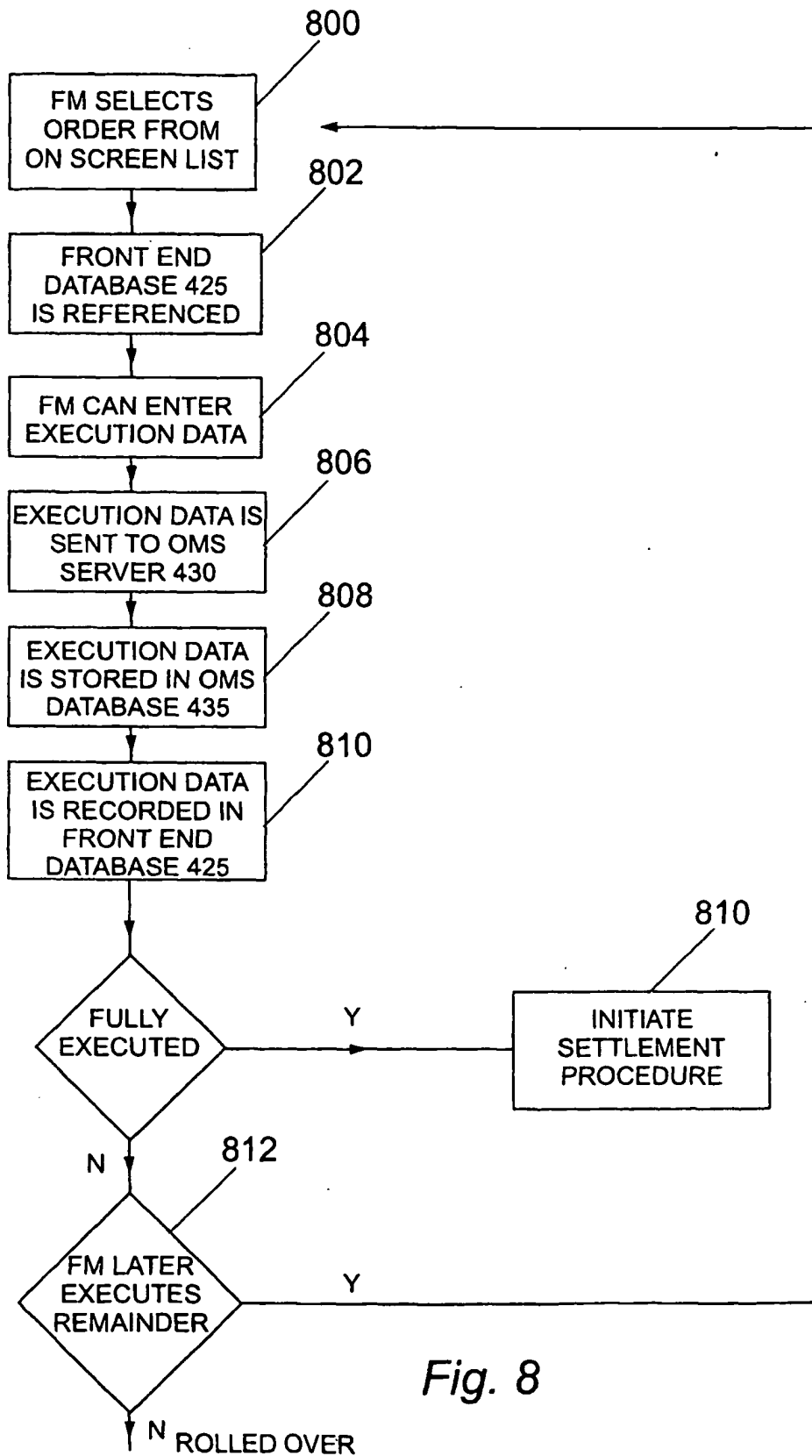


Fig. 8

9/13

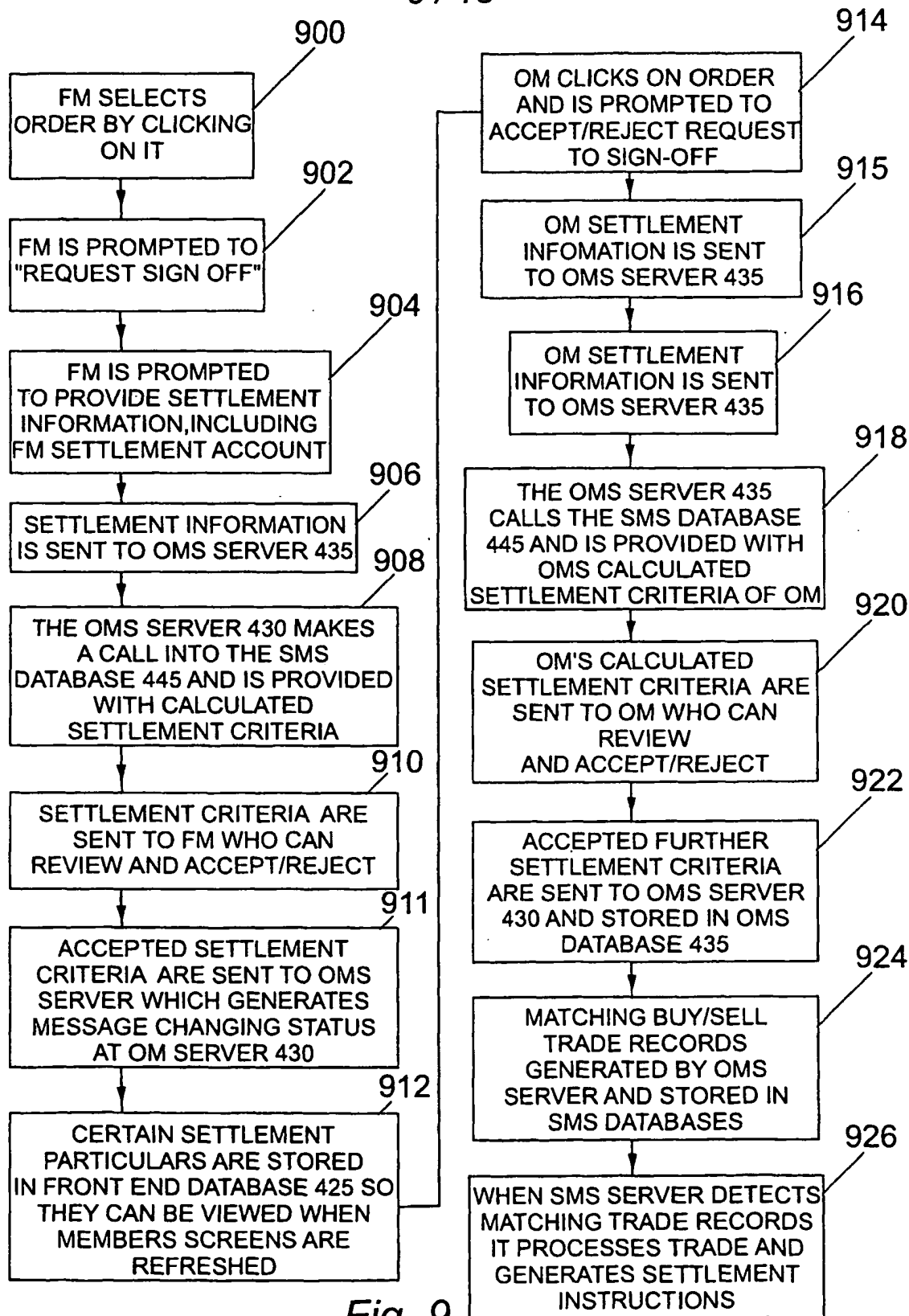


Fig. 9

10 / 13

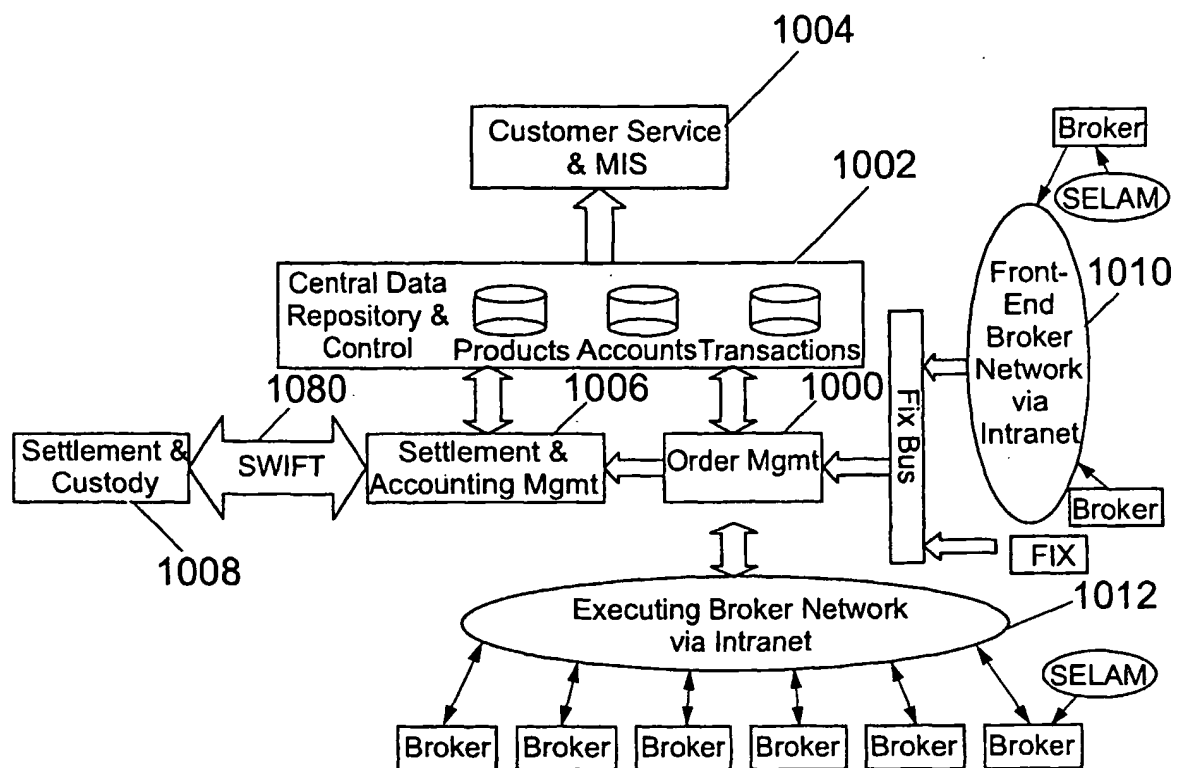


Fig. 10

11 / 13

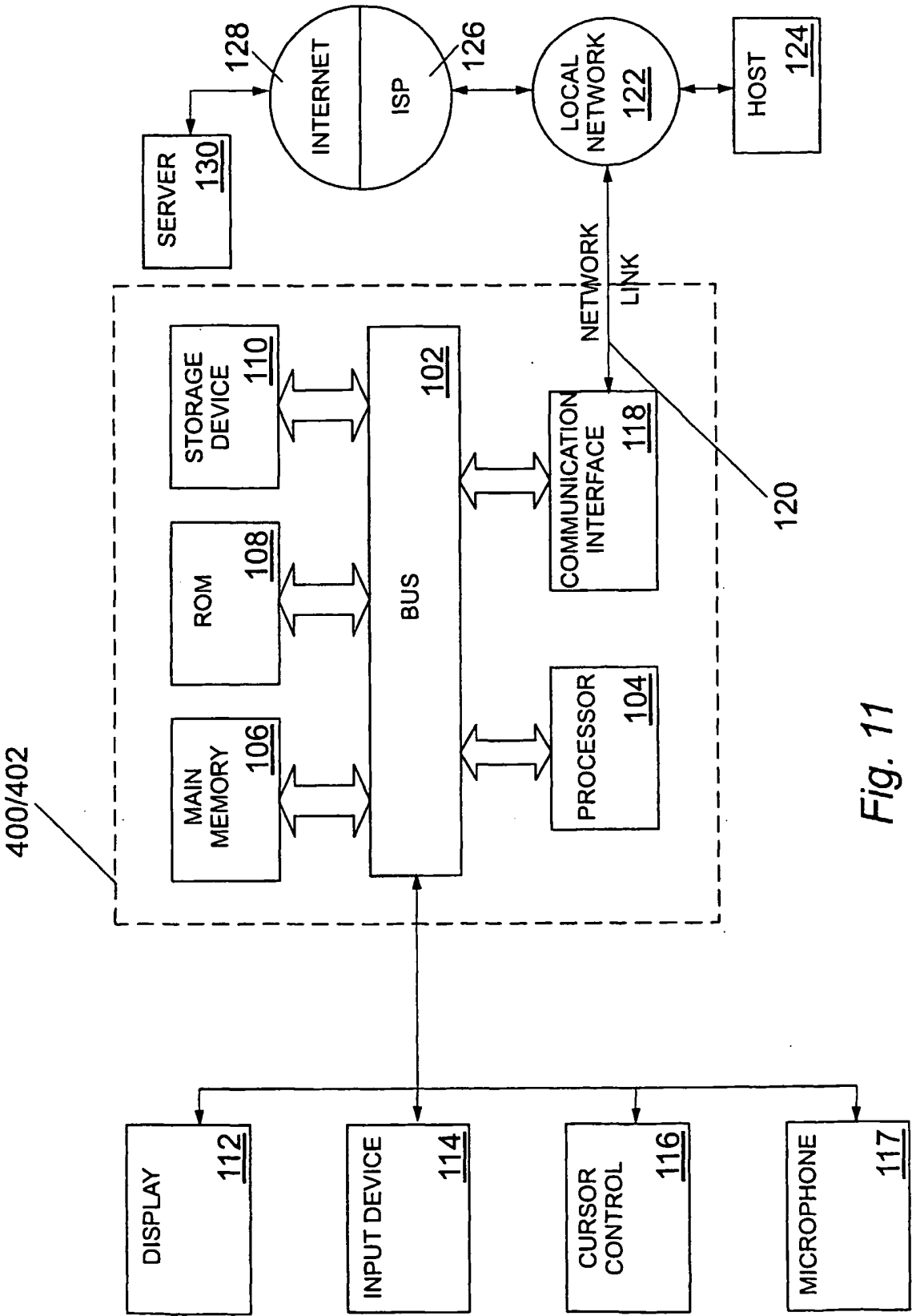


Fig. 11

12 / 13

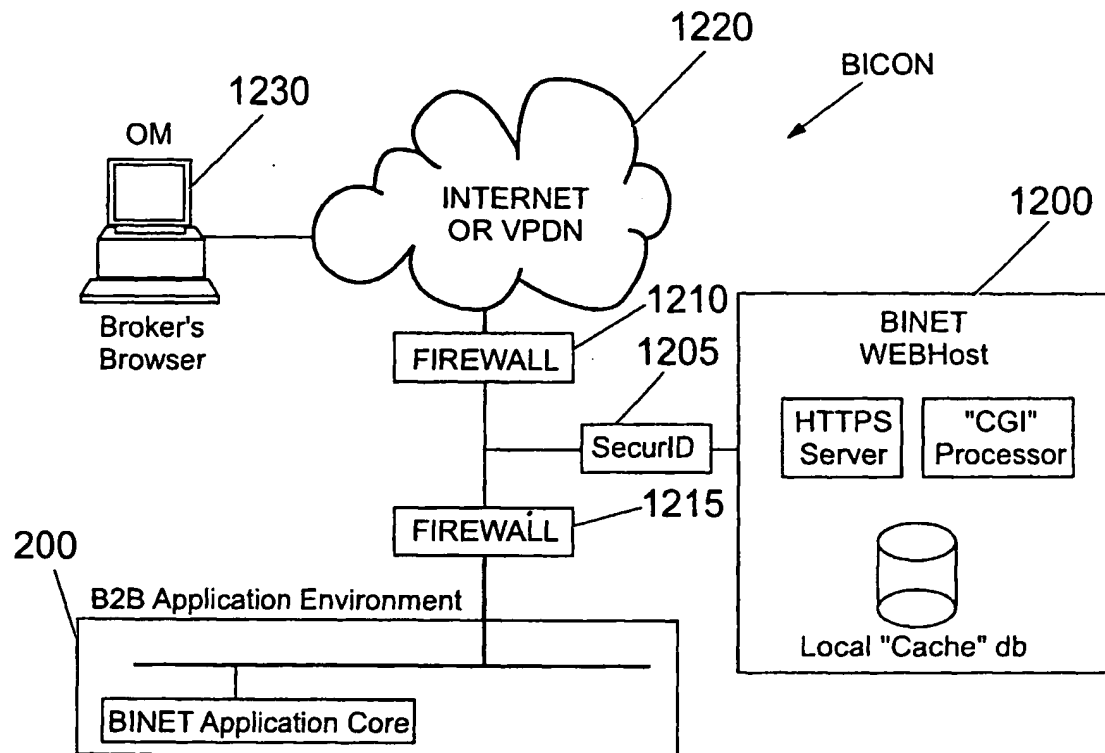


Fig. 12

13 / 13

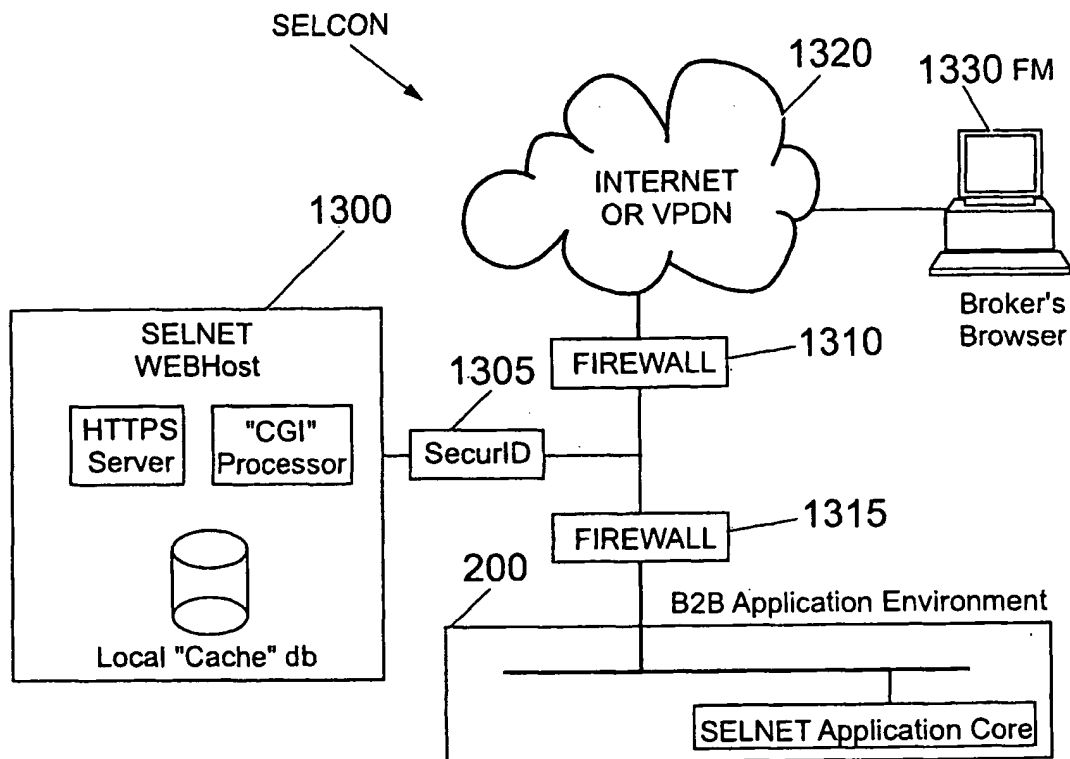


Fig. 13